RTE-IV
Debug Subroutine
Reference Manual

RTE-IV
Debug Subroutine

Reference Manual

**HEWLETT
PACKARD**

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages contain replacement pages or write-in instructions to be merged into the manual by the customer. Manuals will be reprinted as necessary to incorporate all prior updates. A reprinted manual is identical in content (but not in appearance) to the previous edition with all updates incorporated. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change.
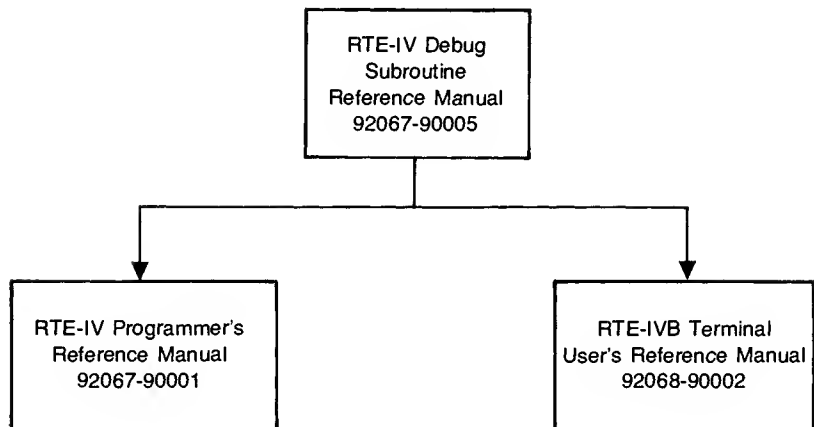
Third Edition .... Feb 1980

DOCUMENTATION MAP

```
                    ┌─────────────────────┐
                    │   RTE-IV Debug      │
                    │    Subroutine       │
                    │  Reference Manual   │
                    │   92067-90005       │
                    └─────────────────────┘
             ┌────────────────┴────────────────┐
             ▼                                  ▼
  ┌─────────────────────┐          ┌─────────────────────┐
  │ RTE-IV Programmer's │          │   RTE-IVB Terminal  │
  │  Reference Manual   │          │ User's Reference Manual │
  │   92067-90001       │          │   92068-90002       │
  └─────────────────────┘          └─────────────────────┘
```

See the *RTE-IV Programmer's Reference Manual* and
RTE-IVB Terminal User's Reference Manual for
complete documentation maps.

CONTENTS

CONTENTS (Continued)

ILLUSTRATIONS

DBUGR is a utility program for debugging programs run on HP 1000 series computers. It features:

* Symbolic or octal printout

* Symbol definition

* Register examination and change

* Paper tape loading and verification

* Memory search

* Memory clear

* Breakpoints

* Map examination.

1-1. OPERATING ENVIRONMENT

DBUGR runs on any HP 1000 series computer equipped with DMS and teleprinter (or CRT) running the RTE-IV operating system. DBUGR itself is approximately 3.6K words in length including symbol table and excluding the breakpoint table. The user symbol table space is fixed at 50 locations in length. Two locations are required for symbols of one or two characters, while three locations are required for symbols of three to six characters. The breakpoint table, which uses 50 words of memory, consists of 10 breakpoints of 5 words each.

Paper tape operations are configured to dump to LU4 and read from LU5.

1-2. LOADING AND USING DBUGR

To use DBUGR it must be relocated with the program to be debugged. This may be done in one of two ways. The first is to use the DB format in the loader. LOADR will set the primary entry point of the program to DBUGR and save the program's actual start address in DBUGR. For example:

        *RU,LOADR,,%PROG,,DB

Note that this is the only way that segment breakpoint can be utilized.

The second method is for the program to call DBUGR directly by the following calling sequence:

```
         Assembly                        Fortran

         EXT  DBUGR                      CALL  DBUGR(lu#-optional
         JSB  DBUGR                                 parameter)
         DEF  RTN
         DEF  lu# of console (optional)
     RTN EQU  *
```

When the user's program begins execution, DBUGR takes control and prints START DBUGR on the optional console logical unit or, if not included, on the logical unit passed to DBUGR through the system subroutine LOGLU. At this point the user can initiate a debug operation. All debug operations are conducted at the Assembly Language level. A load map of the program is essential. If debugging a program written in a higher level language, a mixed listing of source and assembly is required.

DBUGR has 10 breakpoints available which reside in a relocatable module named %SGBPT. If the user desires to change the number of available breakpoints, the user may write his own SGBPT module and load the new code in place of the system library routine SGBPT. This can be done by using the loader LI,%SGBPT command to search %SGBPT file before the system library.

The routine looks like:

```
ASMB,R,Q
         ENT  SGBPT,SGBPE
SGBPT    DEF  *+1               start of table points
         REP  10                determines number of breakpoints (10)
         OCT  0,0,0,0,0
SGBPE    DEF  *                 end of table pointer
         END
```

1-3. ELEMENTS OF DBUGR AND TEXT CONVENTIONS

Input to DBUGR consists of controls and expressions.

1-4. CONTROLS

Controls, consisting of special characters and letters preceded with escape or ALTMODE, act as directives to DBUGR. In the text that follows, escape will be denoted by \, carriage return by CR, control by CTRL, tab by TAB and line feed by LF. (CTRL-J is optional LF on 264x terminals.) Escape prints as a backslash (\) on a TTY or CRT. Some controls are:

```
/
!
\S
=
+
```

For muiltipoint terminals or terminals using driver DVR07, refer to Appendix D.

<div align="center">NOTE</div>

> Some terminals do not print the first backslash after an excape (e.g., the 2640). The control "\U" will cause DBUGR to print two backslashes for each escape.

1-5.   EXPRESSIONS

An expression, which will be denoted in the text by the letter n, consists of one or more terms, combined by operators as in the following:

        AA+10

1-6.   TERMS

A term may be a symbol (denoted in the text by the letter s), a number, or a special notation called a mark. The following are terms:

BB   A symbol is defined as a letter or a period followed by any number of letters, digits or periods. Only the first six characters are significant. For example:

ABC	A5	.B
A.ZZZZZ	.4	...

         Each symbol may be assigned a value. The symbol and its value are equated by entries in the DBUGR symbol table. Values for symbols lie in the range of -32768 through 32767.

3775   An octal number.

386.   A decimal number. The decimal point is used to characterize the number as decimal rather than octal. A decimal point in a number other than as the last character is not meaningful.

. and * Period and Asterisk are marks. When used as terms in an expression their value is equal to that of the current value of DBUGR's location counter.

\Q   Esc-Q is a mark. It refers to the last quantity typed.

1-7.   OPERATORS

Legal operators are:

+   Add terms

Space   Add terms (this is manually faster than shifting and depressing the + key.)

-        Subtract term

,   Inclusive OR terms (inclusive OR is used to selectively set individual bits within a word).

Operators are discussed in more detail in Appendix A.

1-8.   CONVENTIONS FOLLOWED IN THE TEXT

1. What you input to DBUGR is underlined.

2. Use of a control that does not cause print-out is denoted by brackets ([]). These will not be underlined.

3. All values referenced in the text are octal unless otherwise noted.

4. Vertical and horizontal spacing shown in the examples is not exactly as it appears on a teletype or CRT printout.

1-9.   LIMITATIONS

DBUGR reads and interprets each character as it is entered so that many commands may be invoked by a single keystroke. Since one character is read at a time, it is possible for the user to get ahead of DBUGR and get a system prompt. When this happens the user should increase the priority of the program being debugged with the PR command. Alternatively, through use of the \\U command, DBUGR can accept a continuous string or line of characters as would be sent from a 264x terminal by a soft key, cartridge tape unit, or a Multipoint 264x terminal using driver DVR07.

Information can be displayed to us in one of four modes:

1. Symbolic instructions

2. Numeric constants

3. Address pointers

4. ASCII characters

Although a printing mode normally remains in effect until changed, it is possible to either:

a. momentarily invoke a different mode for the display of only one number, or

b. temporarily invoke a different mode for a series of examinations until a carriage return is entered.

2-1.   PRINTOUT AS SYMBOLIC INSTRUCTIONS

DBUGR is in symbolic mode when it is loaded.

In symbolic mode the contents of a memory location are printed as symbolic instructions; the operation codes are represented as mnemonics. The address field of one-word memory reference instructions will be printed as octal numbers with the page number merged with the page offset. Only the first word of multi-word instructions will be interpreted.

It is important to note that in symbolic mode every location is printed as a symbolic instruction whether that location contains an instruction, data or an address. The bit pattern in memory is simply interpreted as an instruction. For example, the octal value 1767 would appear in memory as:

```
bit 15                                   6                         0
    +-------------------------------------------------------------+
    |                                                             |
    | 0   0   0   0   0   1   0   0   1   1    1   1   1   0   0   0 |
    |                                                             |
    +-------------------------------------------------------------+
```

Notice that bits 6 through 15 correspond to the bit pattern for the ALF instruction. The lower 6 bits are similarly treated as instructions. The printout appears as:

    ALF,CLE,ALF

Also notice that not all shift rotate instructions are defined. In those cases DBUGR will print the valid opcodes merged with the remaining bits. For example, the octal value 1747 will be printed as ALF,CLE,7.

DBUGR's symbol table does not contain instructions referencing the overflow register, therefore overflow instructions will be printed as I/O instructions with a select code of 1.

The operand field for I/O and double shift instructions will be printed correctly except when the operand is zero, in which case a blank is printed instead of a zero. Therefore STF 0 will be printed as STF, and RRL 16 (actual operand is 0) will be printed as RRL.

"I" and "C" are special symbols in DBUGR's symbol table with the octal values of 100000 and 1000 respectively. Therefore 100000 and 1000 will be reverse assembled as I and C.

From this discussion, it is obvious that we must know whether we are looking at data or instructions. It is also obvious that it is desirable to be able to print data or address words as octal numbers. DBUGR provides this capability.

2-2.   PRINTOUT AS NUMERIC CONSTANTS

We can either momentarily change the printing mode to octal so the last item printed will be reprinted as octal, with subsequent printouts continuing in the master mode; or we can change the master mode to octal so that all subsequent printouts will be in octal.

Conversely, if we change the master mode to octal, we have the capability of momentarily switching back to symbolic; it works both ways. The examples below demonstrate printing modes. In these examples the following controls are used:

n/   Prints the contents of memory location n. These contents can be modified at this point, but that is discussed later.

\C   Set master print mode to numeric constant. Remember, C means constant.

=   Print the last quantity (typed by either us or DBUGR) as a numeric constant. This is a momentary mode change.

\=   Same as =, except constant mode remains in effect until a carriage return. This is a temporary mode change.

\S   Set master print mode to symbolic instruction. Remember, S means symbolic.

!   Print the last quantity typed as a symbolic instruction. This is a momentary mode change.

\!   Same as !, except symbolic mode remains in effect until a carriage return. This is a temporary mode change.

2-3. CHANGING THE RADIX OF NUMERIC PRINTOUT

An interesting feature of DBUGR is that the numeric output (in any mode) does not have to be octal. The radix of output values can be from 2 to 33. Thus, it is possible to print data as a decimal value or examine memory in hexadecimal. If the radix is decimal, numbers are followed by a period. You can change the output radix by using the control. DBUGR interprets all values as 16- bit unsigned numbers, regardless of radix.

n\R   where n can be any decimal value from 2 to 33 (or octal value from 2 to 41. Remember, R means radix.

Output Modes

For example:

```
    2420/  ADA 2745   \C [LF]                    Select numeric constant
    -----            --                          master mode
    2421/  3004 2\R   [LF]                        Select binary output
                ---
    10100010010/  10000010000 10\R [CR]           Select octal output*
                              ----
    \S   [LF]                                      Select symbolic master
    --                                            mode
    2423/  JMP 2714,I   [LF]

    2424/  SZA   [LF]

    2425/  JMP 2714,I   2\R   [LF]                 Select binary output
                        ---
                •

                •

                •

                •
```

*The radix could have been specified in decimal as 8.\R.

```
    10100010110/ JMP 10111001101,I 20\R [CR]  Select hexadecimal output
                                    ----
    \S [LF]                                        Select symbolic master
    --                                            mode
    517/ JSB 331,I [LF]
    518/ CLA,SSA,SLA,SZA   \C [LF]                 Select constant master
                          --                      mode
    519/   51A   [LF]
    51A/   9B3E 10\R   [CR]                        Select octal output
                ----
    2471/   3004   10.\R [LF]                      Select decimal output
    -----          -----
    1338./ 1040.   [LF]
    1339./  44464.   [LF]
    1340./ 1026. [LF]
```

2-4. ALPHANUMERIC (ASCII) PRINTOUT

It may be necessary to examine alphanumeric information. If so, DBUGR allows us to change the printing mode so that the contents of each memory location are printed as two ASCII characters. In this mode DBUGR will interpret the 16 bits as two 8-bit ASCII codes, whether the memory location contains data, an instruction, or an address. The controls are:

\H   Sets master printing mode to ASCII characters.

'   Prints the last quantity typed as two ASCII characters. Then prints a double quote.

\'   Same as ', except ASCII mode remains in effect until a carriage return.

Study the following example carefully:

```
2514/ 115762              \H [LF]   Select ASCII printout
-----                     --
2515/PV"=50126            ' PV" [LF]
  -                       -
2516/NE"=47105            ' NE" [LF]
  -                       -
2517/@."=40056            ' @." [LF]
  -                       -
2520/-"=2655              ' -"  [LF]
  -                       -
2521/*"=115452            ' *"  [LF]
  -                       -
2522/S"=2523              ' S"  [LF]
  -                       -
2523/-0"=126660           '-0"  [LF]
  -                       -
2524/ >"=115476           '>"   [LF]
  -                       -
2525/+0"=25460            '+0"  [LF]
  -                       -

 .
 .
 .
2532/-"=2655              '-"   [LF]
  -                       -
2533/ "=12                '"    [LF]
  -                       -
2534/ "=5                 '"    [LF]
  -                       -
2535/ *"=115452           '*"   [CR]
  -                       -
```

From an examination of the octal equivalents of the printed characters, several observations can be made:

1. A character will be printed only if the upper and/or lower 8 bits of a word contain bit patterns that represent legitimate ASCII characters.

2. If either 8-bit field does not represent an ASCII character, it is not printed. There is no way to determine whether a single printed character fell in the upper or lower half of the word without looking at the octal equivalent. At location 2522 above, only the lower half of the word converts to the character S.

3. If neither half word contains an ASCII code, but does contain binary information, nothing is printed (location 2534 above). If a non-printing ASCII character is encountered, DBUGR performs the function that ASCII code represents. For example, encountering a 007 causes the bell to sound, and encountering a 012 causes a line-feed.

2-5.   ADDRESS POINTERS

There is one more printing mode, It allows DBUGR to print a 16-bit memory location as an address pointer. Keep in mind that in HP 1000 computer programming, many words are set up to contain a full 15-bit address plus a bit to indicate direct or indirect use of that address. In the following code:

```
        3000          LDA          POINT,I
                       .
                       .
                       .
        3010 POINT    DEF          DATA
                       .
                       .
                       .
        7000 DATA     BSS          1000
```

the DEF pseudo operation generates a full 15-bit address at location 3010. The address generated is that of the symbol DATA; i.e., 7000, and this address is used as a pointer by instructions such as the LDA at 3000 to access the data stored at DATA. The contents of location 3010 look like:

```
bit 15                                                    0
    +------------------------------------------------+
    |0  0  0  0  1  1  1  0  0  0  0  0  0  0  0  0| =7000
    +------------------------------------------------+            8
```

Memory locations containing address words can easily be examined in numeric constant mode. In fact, if we switch to the printing mode that allows us to print the contents of memory locations as addresses, there appears to be no difference in output. The example below demonstrates the apparent identity. The control <- (left arrow or underline on some CRTs) is used to print the last quantity typed in address pointer mode.

```
    \C       [CR]
    --
    1510/   60247    <- 60247    [LF]
    -----            --
    1511/   3004     <- 3004     [LF]
                     --
    1512/   102000   <- 2000,I   [LF]
                     --
```

The two modes return almost identical results! What then is the use of address pointer mode?

Printing in address pointer mode is useful only in conjunction with another feature of DBUGR; i.e., the capability of associating symbols with numeric addresses.

2-6.   DEFINING SYMBOLIC ADDRESSES IN DBUGR

Remember that when a program is in memory for execution, all symbols associated with program locations at assembly time have been converted to absolute addresses; they are gone! As DBUGR maintains its own symbol table, we can associate arbitrary symbols with specific locations. This is useful because we may wish to see locations and address references printed as symbols rather than words.

Before we see how symbols are manipulated, it is necessary to learn how DBUGR sequences through a program. For example, to begin debugging at location 2500, we would type:

    2500/

DBUGR will respond by typing the contents of location 2500 (assume we are in numeric constant print mode). The output would look like:

    2500/ 171572

If we strike LF or CTRL-J on a 264x, the next sequential location in memory is displayed:

    2500/ 171572 [LF]

    2501/   723 [LF]
    2502/   5241

Although it was not explicitly mentioned, in all previous examples advancing to the next line of print was accomplished by LF.

Obviously, DBUGR maintains a location counter similar to the P-register of the computer, and LF is the control for sequentially examining memory. Later we will see that the contents of the memory location just printed can be changed before executing the LF.

If we wish to associate a symbol with the address 2500, we can type the following:

    n<s: where n is an expression resulting in a 16-bit value and s
          is a symbol. The symbol is then said to be defined.

Thus, "less than" (<) is a control to designate a value, and "colon" (:) is a control to define a symbol.

    2500<SYM:

will associate the symbol SYM with the address 2500.

In the following example, notice how symbols replace numeric notation
after location 2500. Assume we are in numeric constant mode, and SYM
has been defined as above.

```
2476/     126720     [LF]
-----
2477/     115476     [LF]
SYM/      2734       [LF]
SYM+1/    115471     [LF]
SYM+2/    25460      [LF]
SYM+3/    126666     [LF]
          .
          .
          .
          .
          .
          .
          .
```

A strange thing happens after location SYM+10:

```
SYM+10/   165715     [LF]
2511/     7004       [LF]
2512/     77113
```

Addresses are associated relative to a particular symbol for 10 octal
locations. If we wish to continue using symbolic addressing, we have
to define another symbol:

```
SYM+130/      165715 [LF]
--------
2631/ 7004   [CR]
2631/ SYM2:  [LF]
-----------
SYM2+1/ 77113
```

Notice the use of CR above. Unlike LF it only returns us to the next
print line; the location counter does not change. Actually it is
generally not necessary to strike a CR before typing a new address or
using a new control. You can continue on the same line.

The exception to the 10 rule above is for symbols of one or two
characters. For these symbols, the numeric offset has no limit.
Thus:

```
2631<S:   [CR]
-------
S+5000/ 7004 [LF]
-------
S+5001/ 165715 [LF]
    .
    .
    .
```

This mode is useful with relocatable code where we might define a
symbol for the origin of each module. The offsets would then be the
same as those printed on the assembly listings.

Another form of symbol definition that would have been more efficient
at location 2631 above is illustrated below:

```
SYM+130/    165715    [LF]
--------
2631/       7004   SYM2:    [LF]
                    -----
SYM2+1/     77113
```

When no value is explicitly designated for a symbol definition, the
current value of the location counter is used.

Now let us look more closely at the printout in symbolic instruction
mode (Column A of Figure 2-1). Notice that we may specify the
address at which to begin printing by using its symbolic form:

A SYMBOLIC INSTRUCTION	B ADDRESS POINTER	C OCTAL CONSTANT
SYM+1/ JSB 1471,I	<-15471,I	=115471
SYM+2/ JMP 1460	<-25460	=25460
SYM+3/ JMP SY+35,I	<-26666,I	=126666
SYM+4/ JSB 1454,I	<-15454,I	=115454
SYM+5/ CLA,CLE,INA,SZA,RSS	<-SYM+7	=2507
SYM+6/ CLE,SEZ,SSA	<-2120	=2120
SYM+7/ JMP 2716,I	<-26716,I	=126716
SYM+10/ JSB 1453,I	<-15453,I	=115453
SYM+11/ CLA,CLE,SSA,SZA,RSS	<-SY+13	=2513
SYM+12/ CLE,SEZ,SSA	<-2120	=2120
SYM+13/ JMP 2716,I	<-26716,I	=126716
SYM+14/ JSB 1762,I	<-15762,I	=115762
SYM+15/ CLA,CLE,SSA,RSS	<-SY+21	=2521
SYM+16/ 2746	<-SY+115	=2746

Figure 2-1.COMPARISON OF SYMBOLIC, ADDRESS POINTER, AND OCTAL PRINTOUT

Notice that not only are the locations printed in symbolic form, but some addresses referenced by memory reference instructions also appear using the offset limits; i.e., 10 for multicharacter symbols or no limit (but >0) for one or two character symbols.

Referring to the octal printout in Column C, notice that the entire word is printed as an octal constant, even if that constant represents an address that can be represented symbolically, as at location SYM+7.

In symbolic instruction mode, symbolic addresses appear only if that word is a one-word memory reference instruction. Address pointers will not be printed symbolically. They will appear either as octal constants, as locations SYM+16 or as meaningless instructions as at location SYM+11.

A glance at Column B in Figure 2-1 immediately reveals that address pointer mode is the only mode that will print 15-bit addresses in symbolic form. Notice that the control <- is used to momentarily invoke address pointer mode. To set the master printing mode to address pointer, use the control:

> \A Sets the master printing mode to address pointer. Remember A means address.

The control \ <- can be used to temporarily invoke address pointer mode until a carriage return is executed.

2-7.   DELETING SYMBOLS FROM THE SYMBOL TABLE

Using the control \K will kill all symbol definitions that we have defined. Symbols defined by DBUGR remain. Remember K means kill.

CAUTION

Do not use the following to define address labels:

1. Instruction mnemonics; if a symbol is redefined, the original value is lost.

2. The letters C and I; these are defined as 1000 and 100000 octal, respectively.

3. Special Register Symbols.

2-8.   ASSIGNING A SYMBOL TO AN ADDRESS JUST PRINTED BY DBUGR

In addition to being able to define a symbol equal in value to a specific address or to the current value of the location counter, we can equate a symbol equal in value to an address just printed by DBUGR:

> s&   Defines symbol s equal in value to the address of the last quantity typed, if an instruction.

For example:

```
1476/    JMP 305 XFER&    [CR]
-----
305/     SLA+17           [CR]    The symbol XFER is defined
----                              as location 305
XFER/    SLA+17           [LF]
-----
XFER+1/ SLA+20
```

2-9.   HOW TO PRINT THE LAST QUANTITY TYPED IN MASTER MODE IF YOU HAVE JUST CHANGED MODES

> ; Prints last quantity typed in master mode.

For example:

```
1700/ HLT 77 =102077 ;HLT 77 [CR]
-----          -         -
1777/ 40502 'AB" ;40502
-----       -    -
```

3-1. CONTROLS TO EXAMINE MEMORY ONLY — LOCATION COUNTER DOES NOT CHANGE

We have already used two controls to examine memory locations:

> n/   To open and print the contents of location n.

> LF   To open and print the contents of the next sequential location.

Actually these controls do more than print the contents of the memory location; they "open" the location for possible modification. Before we explore that possibility, let us look at some other controls for examining memory:

> /   Opens and prints the contents of the location pointed to by the last quantity typed. The quantity typed is used as a 15-bit address, regardless of the printing mode.
>
> For example:
>
> 2007/ 20602 / 67 [LF] Opens and prints contents
> -----   -   of 20602
>
> 2010/ 30501 / 115723   Opens and prints contents
>   -   of 30501
>
> Notice that DBUGR's location counter does not change. We can continue sequential examination of memory simply by using LF.

> \/   Opens and prints the contents of the location pointed to by the last quantity typed. Unlike /, the last quantity typed is interpreted as a memory reference instruction and only the 11-bit address field is used as the effective address. The location counter does not change.

> \L   Prints the contents of the next 16 sequential locations; the last printed location is left open.

> n\L   Same as the \L, the quantity n replacing 16 as the number of locations listed.

For example:

```
5010/     ELA      5\L
-----              ---
5011/     ANA 2042
5012/     CMA,INA
5013/     SSA,RSS
5014/     JMP 3066
5015/     JSB 2076
```

3-2. CONTROLS THAT EXAMINE MEMORY AND SET THE LOCATION COUNTER TO A REMOTE ADDRESS

The following two controls are similar to / and \/, but
notice that the location counter is set to the address of
the memory location being examined.

CTRL-I (TAB)   Opens and prints the contents of the location
(compare   pointed to by the last quantity typed. This
with /)   quantity is used as a 15-bit address pointer.
                   DBUGR's location counter is changed to contain
                   this address.

For example:

```
2312/     2031     [CTRL-I]     Location counter
-----
2031/     126313 [CTRL-I]       change to 2031
26313/    43734                 then to 26313
```

\CTRL-I (\ TAB)   Same as CTRL-I except that the last quantity
(Compare   typed is interpreted as an instruction and
with E/)   only the 11-bit address field is used as
                   the pointer.

For example:

```
5002/   JMP   5007   \   [CTRL-I]   Location counter
-----               -
5007/   SLB+56                      Changes to 5007
```

3-3. CONTROLS TO CHANGE THE CONTENTS OF MEMORY

To change the contents of a memory location, simply open that
location, then type the quantity desired followed by one of the
controls discussed below. The quantity you type is designated by n.

3-4. MEMORY CHANGE CONTROLS THAT MAINTAIN THE LOCATION COUNTER WITHIN THE CURRENT SEQUENCE

nCR   n is the quantity that we want stored in the open
nLF   location. The control that follows this quantity
n^   executes as previously described.

For example:

```
5010/   1600    7777  [CR]   Contents of 5010
-----
*/ 7777                      changed to 7777. Verify by
--                           re-examining 5010.
```

Note that an asterisk is used to imply an address
equal in value to the current value of the location
counter. A period can be used in place of an
asterisk.

```
5012/   27006   6655  [LF]   Contents of 5012
-----           ----
5013/   2060     ^           changed to 6655.  Verify
                 -
5012/   6655          [CR]   by sequencing backward.

5012/   6655    44444  ^     Contents of 5012 changed
-----           -----  -
5011/   7777    [LF]         to octal 44444.  Verify
5012/   44444               by sequencing forward.
```

IMPORTANT: .The use of CR will close the currently open memory
register!

For example:

```
5010/   1600    7777  [CR]   Contents of 5010 becomes
-----           ----
3333    [CR]                 7777.  Attempt to change
----
*/ 7777                      again, but 5010 still con-
--                           tains 7777.
```

NOTE: Note that a period is used above to refer to an address equal
in value to the current value of the location counter.

DBUGR provides a control for storing one or two ASCII characters in a
word. It has the form s" where s is a symbol.

Type the ASCII input and the quote followed by any of the memory change controls discussed above:

```
1700/    20532  AB"      [CR]
-----           ---
*/   40502    'AB"
--            -
```

Only letters, digits, or the period, may be used; blanks and special characters are illegal.

The next two controls allow us to store a new address in an open memory location and then open and examine the contents of the location pointed to by that address.

n%   Stores the quantity n in the currently open memory location; then opens and prints the contents of the location pointed to by n. N is used as a 15-bit address pointer. The location counter does not change.

         Notice that this control is similar to / with the added capability of changing the currently open memory location.

         For example:

```
5011/   7777 55%   102055   [CR]
-----        ---
```

         The contents of 5011 are changed to contain the quantity 55; then location 55 is opened and its contents are printed.

n\%   Same as n% except that n is interpreted as an instruction, and only the 11-bit address field is used as a pointer to the location to be opened and printed.

         For example:

```
4011/ ADB 215 ADB 312\% 4445 [CR]
-----         ----------
312/ 4445 [CR]
----
4001/ ADB 312
-----
```

         The contents of 4001 are changed to point to location 312. Location 312 is then opened and its contents printed. It is then opened again for comparison. Notice that we reprint location 4001 to verify the change.

3-5. MEMORY CHANGE CONTROLS THAT CHANGE THE LOCATION COUNTER TO A REMOTE SEQUENCE OF CODE

The following controls allow us to store a new address in the currently opened location and then open, print and set the location counter to the address into which we have just stored.

nCTRL-I   Stores n in the currently open register and then opens and prints the contents of the location pointed to by n. Remember that the quantity is interpreted as a 15-bit address. The location counter is set to this address.

For example:

```
337/  STB 1772,I   / ALF+73 =1773   1660 [CTRL-I]
----              -              ----
1660/ ISZ 126 =34126  [CR]
               -
1772/ ELA+60 =1660
-----        -
```

Location 337 contains an indirect reference to location 1772, which contains 1773. Before 1660 CTRL-I is typed, location 1772 is the open location. Typing 1660 CTRL-I changes the contents of 1772 to the value 1660 and then opens and prints the contents of location 1660, with the location counter set to this address.

n\CTRL-I   Same as nCTRL-I except that n is interpreted as an instruction.

For example:

```
330/  JMP   461  JMP   500 \[CTRL-I]
----             -------- -
500/  JMP   207,I [CR]
330/  JMP   500
----
```

Location 330 originally contains a JMP 461. This is changed to a JMP 500, and location 500 is opened, printed and the location counter is set to location 500.

This control is excellent for making in-core patches. For example, suppose it is necessary to insert instructions between locations 3737 and 3740:

```
3737/  ADA   200
3740/  SSA
```

The inserted code must be stored at locations remote from the in-line code, therefore, we must jump to the added instructions from location 3737 and return to location 3740.

Assuming that memory is available at location 3745, we can use DBUGR as follows:

```
3737/   ADA   200   JSB   3745   \[CTRL-I]
-----               ------------
3745/   0   [LF]
3746/   0   ADA 200    [LF]
            --- ---
3747/   Inserted code
            .
            .
            .
            .
37xx/   0   JMP 3745,I
            ----------
```

Note that the ADA 200 was moved to the beginning of the inserted code to make room for the JSB instruction.

3-6. SPECIAL REGISTER MODIFICATION AND EXAMINATION

To examine user or system maps, modify or examine O-, E-, X-, Y-registers, or examine the DMS status as of the last breakpoint, special modes are required.

3-7. SPECIAL REGISTERS

\M [CR] will cause the following two lines to be displayed:

AREG	BREG	XREG	YREG	EOREG	MASK	(CBVAL-CBADDR,I)		CBMASK	=	CBTEST
11	237	11376	5775	2	177777	7	22042	177777		SZA

AREG	A-register
BREG	B-register
XREG	X-register
YREG	Y-register
EOREG	E-, O-registers in bits 1 and 0 respectively
MASK	Search Mask
CBVAL	Conditional Breakpoint Value
CBADDR	Conditional Breakpoint Address
CBMASK	Conditional Breakpoint Mask
CBTEST	Conditional Breakpoint Test Instruction

The special registers are accessible by the symbol names given above, the same way user defined symbols may be accessed. The EOREG holds the extended bit status in bit 1, and the overflow bit status in bit 0. For example, EOREG = 3 means extend and overflow are set. EOREG = 1 means extend register clear and overflow register set.

Two other symbols available to the user are:

WRTLU   The EXEC control word of the device for DBUGR output.
         (Refer to the RTE-IV or RTE-IVB Programmer's Reference
         Manual regarding EXEC calls.)

BRFLG   0 means check for break in debug loops

         1 means no check will be made for break
                 (speeds up loop processing)

3-8.   MEM STATUS SPECIAL MODE

The following special mode displays the MEM status.

\? displays the MEM status as of the last breakpoint in the
   following format:

       MS =   X15 X14 X13 X12 X10 YYYY

where X15, X14, X13, X12, X11 and X10 = MEM status register bits 15,
14, 13, 12, 11, and 10 respectively.   YYYY is the four digit octal
number representing the base page fence.

3-9.   MAP EXAMINATION SPECIAL MODE

The following special mode allows examination of the system and user
maps and cross load.

\J puts DBUGR into this special mode and responds with a CR LF and
    three spaces.

UM Displays the user map.

SM Displays the system map.

XL Sets up this special mode to cross load from an address in the
    alternate map.

PA Displays the port A maps.

PB Displays the port B maps.

A   Aborts the special mode with no change.

UM:
--

```
 0 = MR0      20 = MR20
 1 = MR1      21 = MR21
 2 = MR2      22 = MR22
 3 = MR3      23 = MR23
 4 = MR4      24 = MR24
 5 = MR5      25 = MR25
 6 = MR6      26 = MR26
 7 = MR7      27 = MR27
10 = MR10     30 = MR30
11 = MR11     31 = MR31
12 = MR12     32 = MR32
13 = MR13     33 = MR33
14 = MR14     34 = MR34
15 = MR15     35 = MR35
16 = MR16     36 = MR36
17 = MR17     37 = MR37
```

The system map registers routine operates the same as the user map registers routine except "UM" is replaced by "SM".

The cross load routine begins by outputting a CR, LF and six spaces with XL followed by three more spaces. The operator then enters an address and a slash to display the contents of that address in the alternate map. DBUGR will return to allow additional cross loads.

For example:

XL
--

      XL   ADDRESS/ (old contents)

      XL

An LF will also increment the address counter as in the case of examining memory locations.

3-10. CONTROLS TO TEMPORARILY CHANGE THE PRINT MODE OVER A SERIES OF EXAMINATIONS

There are several controls that we can use in place of n/ and n\/ (refer to paragraph 3-1). As with \!, \=, \' and \<- they temporarily change the print mode until we execute a carriage return, thereby returning to master mode. However, unlike \!, \=, \' and \<-, they open and print the contents of a register.

n$ Same as n/ but sets temporary print mode to symbolic instruction.

n#   Same as n/ but sets temporary print mode to numeric constant.

n@   Same as n/ but sets temporary print mode to address pointer.

n)   Same as n/ but sets temporary print mode to ASCII.

n\$ Same as n\/ but sets temporary print mode to symbolic instruction.

n\#   Same as n\/ but sets temporary print mode to numeric constant.

n\@   Same as n\/ but sets temporary print mode to address pointer.

These controls can be used without a preceding address expression (n), in which case they can be used in place of / and \/. However, the mode change will be momentary only. Below are given two examples.

Example #1:

```
\C                                                    Begin in constant mode
--
2537/   421    [CR]
-----
2540$  LDA   2534,I   / LDA 62572 / 234 [LF]   Change to symbolic
-----                 -            -            temporary mode
2541/  ADA   2740     [LF]
2542/  SSA           [LF]
2543/  JMP   2657     [LF]
2544/  CLA           [LF]
2545/  STA   2663     [LF]
2546/  LDB   2661     [LF]
2547/  STB   2662     [CR]              Return to constant mode
 [LF]
2550/        125503   [LF]
2551/        46664    [LF]
2552/        6020     [CR]
4000$        SZA      [LF]              Change to symbolic
-----                                   temporary mode
4001/  JMP   4016     [LF]
4002/  LDA   1717     [LF]
4003/  STA   5235   / CPA   72525   / JSB 34 [LF]
                    -                -
4004/  STA   5236     [CR]              Return to constant mode
 [LF]
4005/  53266         [LF]
4006/  73237         [LF]
4007/  17575         [LF]
```

Example #2:

```
\S                                                    Begin in symbolic mode
--
4012/  LDB   1717     [LF]
4013/  ADB   5273     [LF]
4014/  CLA           [LF]
4015/  STA   1,I      [LF]
4016/  LDA   4103     [LF]
5000#  2003  /  72053 [LF]              Change to constant
-----        -                          temporary mode
5001/  126775        [LF]
5002/  73240         [LF]
5003/  12306         [LF]
5004/  73242         [LF]              Return to master mode
5005/  2400          [CR]
 [LF]
5006/  STA   1,I      [LF]
5007/  LDA   5240     [LF]
```

CONTROLS TO LOAD, PUNCH AND VERIFY PAPER TAPE

SECTION IV

4-1. PROGRAM LOAD

DBUGR provides a control for loading absolute binary paper tape:

\Y   Loads absolute binary paper tape, as prepared by the HP Assembler. Mount tape to be loaded before typing \Y. Remember, Y means yank. Loading will terminate at end of tape, or upon any attempt to load beyond the upper limit. DBUGR will report an error for any attempt to load over itself.

4-2. PUNCHING BINARY TAPES AND PRODUCING PATCHES

Memory locations can be changed and punched on paper tape one by one, or blocks of memory can be punched.

As the tape is punched in absolute format readable by HP absolute program loaders, we can use these controls to punch entire programs or binary patches.

nl<n2>\D   Punches locations nl and n2 inclusive

n\D   Stores the value n into the open register, if any; then punches the register last opened (with the new contents, if stored).

Let's say we have have discovered a bug and must make the following changes:

Location	Instruction	Change
2050	RBR	RAR
2051	SSB	SSA

Using DBUGR in symbolic instruction mode:

```
2050/  RBR  RAR  \D  [LF]
-----       ---  --
2051/  SSB  SSA  \D
            ---  --
```

One record is punched for each word.

4-1

4-3.   TAPE VERIFICATION

To verify a tape, we use the following control:

\V   Verifies paper tape against memory. Remember V means verify.
      Mount tape prior to typing \V.   Discrepencies are listed as
      follows:

      Address/ [contents of memory] [contents of tape]

      If the word on tape is zero, no discrepancy is listed.
      Verification will terminate at end of tape or upon any
      attempt to verify beyond the upper limit. DBUGR will report
      discrepancies within itself.

MEMORY SEARCH AND CLEAR

5-1. MEMORY SEARCH

We can search memory within specified addresses for words that are either equal or not equal under mask to a specified word.

Actually, memory is searched for a quantity that is equal or not equal to a quantity formed by taking the logical product (AND) between the mask and each memory location examined. This allows us to search for only a specified group of bits within each word.

5-2. THE SEARCH MASK

The mask is stored at a location referenced by the symbol "MASK". Initially it is set to 177777. Now let us open and print location MASK:

        MASK/ 177777   We can now change the mask.

5-3. LOGICAL PRODUCT REVIEWED

If we take the logical product between a mask and another word, we end up with a result that saves the original value (either 0 or 1) of all bits in the word masked by 1's and clears or extracts all bits in the word masked by 0's. Suppose we wish to search on the lower eight bits in a word:

        Mask = 000377   = 0000000011111111
                     8   2
        Word = 073123   = 0111011001010011
                     8   2
               -------   ----------------

    Logical = 000123   = 0000000001010011
    Product   8   2

The result of the operation is to clear out all bits in which we are not interested and to save only the lower eight bits for comparison against some 8-bit value.

5-4.   SEARCH LIMITS

The following controls establish search limits:

   n<   Sets the lower limit to n.

   n>   Sets the upper limit to n.

The limiting addresses are included in the search.

5-5.   EQUALITY SEARCH

   n\W   Searches within the limits specified for all words equal
         under mask to n. Remember, W means word.

         Example #1: Search within locations 2000 and 3000 inclusive
                      for all halt instructions (1020xx).

         MASK/   177777   177700   [LF]   Change mask to ignore
         -----   ------   bits 0-5

         2000<3000>102000\W

         2234/   102002
         2337/   102002
         2361/   102011
         2572/   102002
         2641/   102002
         2710/   102002
         2743/   102002
         2756/   102002

         Example #2: Search   within   locations   77700   and   77777
                      inclusive for all halt instructions in the Basic
                      Binary Loader (BBL).

         MASK/   177700

         77700<77777>102000\W

         77715/   102077
         77740/   102000
         77747/   102011
         77751/   102055

5-6.   INEQUALITY SEARCH

   n\N   Searches within the limits specified for all words under
           mask not equal to n. Remember, N means not equal to word.

For example: Print memory within locations 2270 and 2315 inclusive.
Do not print words equal to zero.

```
MASK/    177777    [CR]          Verify mask
-----
2270<2315>0\N
-------------
2270/    50262
2271/    26303
2272/    26255
2273/    64124
2274/    6002
2275/    26246
2276/    34242
2277/    170155
2311/    70155
2312/    160156
2313/    70243
2314/    30374
2315/    170574
```

We infer that locations 2300 through 2310 contain zero.

5-7. CLEAR OR SET MEMORY

n\\Z Zero, or set memory within the limits specified to n.
Remember, Z means zero.

Example #1: Set memory locations 2300 to 2310 inclusive to 177777
(octal); then print these locations to verify the
operation.

```
2300<2310>177777\\z     [CR]
-------------------
2270<2310>177777\W
-------------------
2300/    177777
2301/    177777
2302/    177777
2303/    177777
2304/    177777
2305/    177777
2306/    177777
2307/    177777
2310/    177777
```

Example #2: Zero locations 2300 to 2310 inclusive.

```
2300<2310>0\\z
--------------
```

5-8.   EFFECTIVE ADDRESS SEARCH

The effective address is the location an instruction must reference
to acquire or store data. Many times in HP1000 programming addresses
are referenced indirectly:

          LDA   1772,I

The above instruction must further reference location 1772 to obtain
an effective address from which to load the A-register. If the
contents of 1772 = 02340, the LDA instruction would acquire data from
location 02340.

        n\E   Finds all instructions within the limits specified that
                   effectively address location n. Indirect chains are
                   followed to a depth of 16. Normally the mask is set to
                   all 1's, but if not, n is treated as specifying a range
                   of   addresses,   and   all   instructions   effectively
                   referencing addresses within that range are printed.
                   Remember, E means effective address.

Example #1: Find all instructions within 2000 and 3000 inclusive
             that reference location 70567.

             MASK/   177777   [CR]

             2000<3000>70567\E

             2050/   165775   2050 references 70567 indirectly
                                          through 1775.

             1775/   70567   Print location 1775 to verify
                                          operation.

Example #2: Find all instructions within 2000 and 3000 inclusive
             that effectively address any location in the range 4060
             to 4067.

             MASK/   177777   177770   [CR]
             -----   ------
             2000<3000>4065\E

             2500/   JMP   1500,I   All instructions indirectly
             2550/   JMP   1503,I   reference locations in the
             2600/   ADA   1503,I   range 4060 to 4067.
             2700/   JMP   1503,I
             \C
             --
             1500/   4060   [LF]   Verify by printing the
             -----   contents of the pointers.
             1501/   4062   [CR]
             1503/   4065

Note the effect of the mask. By placing a 0 as the lowest octal digit of the mask, no specific address is specified for this digit position, therefore the entire range of addresses from 4060 to 4067 is processed.

BREAKPOINT/TRACE DEBUGGING

6-1. INTRODUCTION

Breakpoint trace debugging is simply a matter of forcing the computer to halt at a particular point in its program execution so we can look at what is happening. We can examine operational registers, dump memory and make memory patches. Of course, we must have a means of resuming execution after a breakpoint and the ability to control the number of times we actually break within a programmed loop.

Combining memory examination and change controls, search and print controls, and binary patching with the breakpoint trace capability makes DBUGR the powerful tool that it is.

.To aid in the tracing operations of a program, a breakpoint may be set at most instructions. When control reaches that instruction, it is not immediately executed; DBUGR gains control and prints out the following:

    ADDRESS(INSTRUCTION)   A   B   EO STATUS

1. Program location counter (P-register) (in address mode)

2. The "broken" instruction (in instruction mode)

3. Contents of the A- and B-registers (starts in constant mode may be changed - see Section 3-7)

4. As one word the extend and overflow registers (in constant mode - see Section 3-7).

Segment breakpoints exist for segmented programs with DBUGR appended to them by the RTE-IV loader. Breakpoints may be set in four different modes:

1. ["A]\B   Break at entry to ALL segments

2. ["N]\B   Break at entry to NO segments

3. [xxxxx]\B   Break at entry to a specific segment

4. n[xxxxx]\B   Break at a defined address within a specified segment.

A breakpoint will be set when the specified segment is loaded into memory. Thereafter, breakpoints remain in effect until a new segment is loaded. At that time breakpoints associated with the newly loaded segment will be set. At such time as the previous segment is re-loaded, the breakpoints associated with that segment will be set, and the breakpoints associated with its predecessor will become dormant.

A segment entry breakpoint displayed just after segment load would look like:

```
SEGMENT  XXXXX  BREAK
ADDRESS(INSTRUCTION)   A   B   X   Y    EO STATUS
```

When a segment entry breakpoint is defined, DBUGR makes no check for the validity of the segment name. Therefore, segment names should not begin with ("A) or ("N).

6-2. CONDITIONAL BREAKPOINT

A conditional breakpoint can be set that will only break when a particular memory location is equal (or not equal) to a particular value after being masked. Location CBVAL, CBMASK, CBADDR, and CBTEST are used to set up the conditions required for the breakpoint to occur. The conditions are set up as follows:

CBVAL   Compare Value

CBMASK   Mask Value

CBADDR   Memory Address to be tested (A-register =0, B-register =1)

CBTEST   Condition required for break. For equality enter SZA or 2002; for inequality enter SZA, RSS or 2003. Note that the comparison is logical and not arithmetical.

The conditional breakpoint may be invoked only upon an existing breakpoint. Moreover, the program must be proceeding from that breakpoint. Conditions will thereafter be tested only at that particular breakpoint.

The conditional breakpoint is invoked by the double escape P command (\\P). Each time the breakpoint is encountered, the contents of the address specified by CBADDR will be XOR´ed with the contents of CBVAL. The result will be AND´ed with CBMASK and this final result will be tested with the instruction in CBTEST (SZA or SZA,RSS). The equation for this test appears in the "\M" display:

    (CBVAL-CBADDR,I)^CBMASK = CBTEST

If the test instruction "skips", the conditional breakpoint will "break". If not, DBUGR will execute the instructions and continue. The conditional breakpoint may be counted with the n\\P command. In this case the conditions will be checked before the breakpoint is counted. A count is made only if a break would have been made. Therefore, n\\P will count n breaks before printing a break message.

In order to delete a breakpoint from the breakpoint table, give the \B command. Then the breakpoint table will be listed:

```
0            M+347       breakpoint in main memory
1 SEG1       77777,I     segment entry breakpoint
2 SEG2       BETA        breakpoint in SEG1
3 SEG3       PHI+5       breakpoint in SEG2
ENTER INDEX OF BP TO DELETE, A to END 1 [CR]
                                       - ----
ENTER INDEX OF BP TO DELETE, A to END A [CR]
                                       - ----
```

Breakpoint 1, the segment entry breakpoint for SEG1, was deleted.

A faster command for deleting breakpoints is \\B which deletes all breakpoints at once.

If an attempt is made to enter more breakpoints than the capacity of the breakpoint table, an error message will be displayed.

```
NO MORE ROOM FOR BREAKPOINTS
```

The user should either delete some breakpoints or create a larger breakpoint table module to be loaded with the DBUGR in the future.

6-3. RESTRICTIONS

We are, however, restricted in the instructions at which a breakpoint can be set. This is because DBUGR gains control at the breakpoint by replacing the programmed instruction with a jump to DBUGR itself. After the break has been accomplished, the instruction is either executed from its temporary location within DBUGR or execution resumes at another address we specify.

DO NOT break at the following:

1. Instruction used as a constant.

2. Instruction that is used as an address pointer in an indirect chain of an instruction.

3. Instruction that is program modified (e.g. a configured I/O instruction).

4. EIG or DMS instructions listed in Figure 6-1

CAUTION

In DBUGR, an attempt to JSB to a point below the MP
fence (except in an EXEC call) as the result of a
trace or proceed command causes DBUGR to reject the
command and to print the break message for the
violating instruction followed by "MP?". You can get
around this problem with a trace. If the offending
instruction is no longer a breakpoint, you can
proceed. The solution then is to move the breakpoint
to a point after the system call and then proceed.

```
+------------------------------------------------------+
|                                                      |
|                   Not Allowed                        |
|                                                      |
|         DJP    DJS    SJS    SJU    UJP              |
|         UJS    XCA    XCB    XSA    XSB              |
|                                                      |
+------------------------------------------------------+
```

Figure 6-1.   Untraceable Instructions

6-4.   CONTROLS

Breakpoint controls in DBUGR are:

n\B   Set a breakpoint at n.   Remember, B means breakpoint.

\B   Enter remove breakpoint mode.

\\B   Remove all breakpoints.

["A]\B   Break at entry to ALL segments.

["N]\B   Break at entry of NO segments (remove segment
                 breakpoint).

[SEG1]\B   Break at entry to a segment named SEG1.

n[SEG1]\B   Break at n within a segment named SEG1.

\P   Proceed with program execution after a break.
                 Remember P means proceed.

\\P   Proceed with conditional breakpoint invoked.

n\P   Proceed; do not trap until n breaks from now.   All
                 breakpoints encountered contribute to the count.

\O   Change break point and trace   register print mode to
               the current master print mode.

n\G   Go to location n; begin execution with flags and
               accumulators as they were at break. Remember, G means
               go.

n\X   Execute the instruction n, then return control to
               DBUGR. If the instruction is a jump, DBUGR loses
               control.   DBUGR prints a carriage return/line-feed
               before and after executing the instruction. If the
               instruction   performs   a   skip,   DBUGR prints an
               additional carriage return/line-feed.   Remember, X
               means execute.

\T   Trace   one   line   of code.   DBUGR simulates   the
               instruction printed in the last break message and then
               prints a new break message with the new location,
               instruction and registers. Two-word instructions also
               print   the   address used by the instruction. Tracing
               data is unpredictable.

n\T   Trace n instructions. Causes n break messages to be
               printed.

\\T   Trace   through a subroutine call. Unlike a \T trace of
               a JSB, a one-shot breakpoint is placed at "..+1".
               Control passes to the subroutine and is regained upon
               return with a break. The one-time breakpoint is thus
               removed. This command should not be used when the JSB
               is followed by an argument list. Also, it should not
               be used when the called subroutine may return to
               somewhere other than "..+1".

Example #1: Break at location 77710.

77710\B   Set breakpoint at 77710.

77700\G   Begin execution at 77700.

77710 (CLA) 173775   13 14 0   DBUGR prints this.

EOREG/   0   Examine EOREG for flags.

Breakpoint/trace debugging

Example #2:   Breakpoint within a loop.

```
  LOOP+15\B                            Set breakpoint at LOOP+15.

  LOOP\G                               Begin execution at LOOP.

  LOOP+15 (CLE) 4324  17  0 41  3      DBUGR prints this.

  EOREG/        3                      Examine EOREG for flags.

  10\P                                 Proceed

  LOOP +15 (CLE) 57211 5  2 24  3      DBUGR prints this
  \P                                   Continue execution.
```

Example #3:

```
    *RU,MAIN                              Execute program loaded
                                          with DBUGR
      START DBUGR
    16002<M:        23456<S:    [CR]      Set start address of MAIN and SEG
    --------        --------
    S+5\B                                 Set Breakpoint in Segment AREA
    --------
    \P
    --
    SEGMENT SEG1 BREAK
    S   (0)  17542 5608 17702  22  2
    S+5[SEG2]\B                           Set Breakpoint in SEG2
    -----------
    \P
    --
    SEGMENT SEG2 BREAK
    S+5 (0)  17542 5606 0 45 22
    M+50\B                                Set Breakpoint in MAIN
    ------
    [SEG4]\B                              Set entry Breakpoint in SEG 4
    --------
    \P
    --
    M+50(LDA M+700)  0 2234 54 72 1  Break in MAIN
    \P
    --

    SEGMENT SEG4 BREAK
    S   (0)  17445 5562 7422 3322 1
                                          Break at segment load.
    \P
    --
    S+10 (JSB 112,I)   24 0 177777 55 2
    ["N]\B                                Clear segment Breakpoint except
    ------                                for specified segments.
    \P
    --
    END DBUGR
```

Example #4:

:RU,PROG	Execute program loaded with DBUGR
START DBUGR	
34221\B	Set breakpoint
\P	Proceed to breakpoint
34221(LDA 1) 43 243 0 77 1	
CBVAL/ 0 130[LF]	Set compare value
CBMASK/ 177777 377[LF]	Set mask to lower byte
CBADDR/ 0 1 [LF]	Test address (B-register)
CBTEST/ SZA [CR]	Test for equality
\\P	Proceed with conditional break invoked
34221 (LDA 1) 32562	23530 2372 0 0
	The lower byte of B-register is equal to test value.
1/23530 23400 [CR]	Change lower byte of B-register to 0
\\P	Proceed with conditional break invoked
34221 (LDA 1) 177777 730 2372 0 0	
\P	Proceed without conditions
34221 (LDA 1) 44 25 3372 0 0	
\P	
END DBUGR	

6-5. MAGIC SYMBOLS

DBUGR has two magic symbols "." and "..". A magic symbol is one of
DBUGR's working registers (variables) that is also in its symbol
table.   We already know about "." which is DBUGR's location counter.
Normally "." is set to some value with one of the many commands that
open a register, but it also may be set this way:

```
201 < .:
--------
.= 201     [LF]    verify "." was set
------
202/  CLA
```

".." is defined, whenever DBUGR is entered or processing a break
point message, to be the next instruction's address. Thus if we have
just broken:

```
3011  (CLA)    10    4010    12  77  2     ..=  3011
                                           ---
```

".." is move each trace and/or break. This provides a convenient way
of opening the broken location but is even more powerful when we use
it to move the trace location. Thus:

```
3011  (CLA)    10    4010    12  77  2 [CR]

..-1/  STA    3040  JMP 3022 [CR] patch previous instruction
-----         --------

.<..:  \T                         set ".." to current location
-----  --                         and trace one instruction

3022  (LDA)  3045 10  4010   12  77  2    (after the jump)
```

Now that we have used most of the controls in DBUGR let's take a closer look at the use of operators (refer to Section 1-7.) within address expressions.

A-1. PLUS, BLANK AND MINUS

The minus sign (-) performs subtraction. The operators plus (+) or blank are used to perform addition. Suppose we wish to change the jump at location 2513:

```
    2513/   JMP   2522   JMP 2522+3     Type  jump address as a relative
    -----                ----------     address      and     verify    by
                                        reexamining 2513.
    ./      JMP   2525
    --
```

Or, let's print the contents of SYM+5, a symbol we have defined as being equal to 2500:

```
    SYM+5/   INA        Type  the symbolic address  and verify by printing
    ------               the  contents  of  the  same address  expressed in
                         octal.
    2505/    INA
    -----
```

A-2. INCLUSIVE OR

Now we use the operator comma (,) to inclusive OR a CLA with an INA,
thereby creating the combined register operation CLA, INA:

```
62572/   CLA     CLA,INA   [CR]    Change register 62572.
------           -------

  .      2404                      Verify new contents in octal.
 --

or in octal:

\C
--

62572/   2400    002400,002004
------           -------------
```

A-3. THE MARK \Q

Note that in the previous example we had to retype the CLA to use it
in the inclusive OR expression. This is not really a lot of trouble,
but we do have a mark that, when in an expression, implies the last
quantity typed. Thus,

```
62572/   CLA     CLA,INA           can also be accomplished by the
------           -------           mark, EQ:

62572/   CLA     \Q,INA
------           ------

 ./      CLA,INA    =2404
 --               -
```

The mark \Q implies the last quantity typed, in this instance CLA.
Remember that Q means last quantity typed.

For example:

```
205U/    2004    \Q+\Q  = 4010   Q+100000 = 104010
-----            -----  -        -------- -
```

A-4. USING DBUGR TO DO SIMPLE ARITHMETIC

We can use DBUGR to evaluate simple expressions, and to convert
decimal or octal numbers to other radices. The following sections
illustrate this capability.

A-5. OCTAL ADDITION AND SUBTRACTION

```
[CR]
234+766=1222
--------

5444-45=5377
--------

4564+567-34+125=5444
----------------
```

A-6. OTHER CONVERSIONS

```
2.\R          Change to binary output
----

23=1001       Convert octal to binary
---

16\ R         Change to hexadecimal output
-----

14=C          Convert octal to hexadecimal
----

78.=4E        Convert decimal to hexadecimal
----
```

A-7. DECIMAL ADDITION AND SUBTRACTION

```
10.\R
-----

45.+56.=101.
--------

578.+9788.=10366.
-----------

400.-45.-26.-78.=251.
-----------------
```

A-8.   OCTAL TO DECIMAL CONVERSION

10.\R

245=165.

256+77+654-147=562.

A-9.   DECIMAL TO OCTAL CONVERSION

89.=131

7859.=17263

78.+59.=211

998.-997.=1

DBUGR will recognize various types of errors. The messages and their meanings are as follows:

X
   You pressed the RUB OUT key to delete a typing mistake. DBUGR will ignore any prior partial expression.

?
   You used an unassigned control. Any prior expression is ignored. Input Error in special mode.

U
   The symbol last used was undefined and a definition was required. The entire preceding expression is ignored.

P?
   Page error: You caused a memory reference instruction to reference an address not in the current page or the base page. The expression is ignored. DBUGR's conception of the "current page" can be changed by examining any location in the desired page.

CHK
   A checksum error has occurred.

MP?
   DBUGR detected a possibly legal instruction that it cannot trace (or proceed with the break point at) without violating memory protect. Move the break point and proceed.

IN?
   An instruction that is legal in the 21XX base set but not executable by DBUGR was detected and DBUGR cannot trace (or proceed with the breakpoint at). Move the breakpoint and proceed. User attempted to set a breakpoint on an instruction DBUGR cannot proceed from. Execution of an instruction using the (instruction)EX feature of DBUGR was attempted using a two or more word instruction (not supported - only one word instructions can be executed using this feature).

TP?
   An attempt to trace, set breakpoint, or paper tape load into DBUGR. Loading, tracing, or setting of breakpoint is terminated.

O?
   Symbol table overflow.

DM?
   An attempt to access memory that is beyond the background partition.

Table C-1. Quick Reference To Frequently Used Commands

```
+-------------------------------------------------------------------+
|   COMMAND         EXPLANATION                                      |
|                                                                   |
|   n/              print contents of location n                    |
|   \L              list 16 locations                               |
|   n\L             list n locations                                |
|   n\R             set Radix to n                                  |
|   \S              Symbolic mode                                   |
|   \C              Constant mode                                   |
|   \4              hollerith mode                                  |
|   \A              address mode                                    |
|   n<sym:          define sym as n                                 |
|   n\X             execute n                                       |
|   n\G             Go to n                                         |
|   \P              proceed                                         |
|   \\P             proceed with conditional breakpoint             |
|   \T              trace an instruction                            |
|   n\P             proceed for n breaks                            |
|   n\\P            proceed with cond. breakpoint for n breaks      |
|   n\T             trace n instructions                            |
|   n\B             set breakpoint at n                             |
|   [x]\B           set entry breakpoint for seg x                  |
|   n[x]\B          set breakpoint in seg x at n                    |
|   \\B             clear all breakpoints                           |
|   nl<n2>n3\W      search nl to n2 for n3 under mask               |
|   nl<n2>n3\N      search nl to n2 for not equal to n3 under mask  |
|   \M              display special register                        |
|   \J              display map register                           |
+-------------------------------------------------------------------+
```

C-1. MODE CONTROL

DBUGR has several basic printing modes:

- As symbolic instructions. This is the mode DBUGR is in when loaded. Controls are:

  \S   Set the master printing mode to symbolic instruction.

  !   Print the last quantity typed as an instruction.

  \!   Set temporary printing mode to instruction, and print the last quantity typed as an instruction.

- As address pointer.

    \A   Set the master printing mode to address. If only initial symbols are defined, this is equivalent to constant.

    <-   Print the last quantity typed as an address pointer.

    \<-   Set temporary printing mode to address, and print last quantity typed as an address.

- As constants, in a specified radix.

    \C   Set the master printing mode to constants, in the current radix.

    =   Print the last quantity typed as a constant.

    \=   Set the temporary printing mode to constant, and print the last quantity typed as a constant.

    n\R   Set the output radix to n.

- As ASCII characters in halfwords.

    \H   Set the master printing mode to ASCII characters in half-words.

    '   Print the last quantity typed as two ASCII characters. Then print ".

    \'   Set the temporary printing mode to ASCII, and print the last quantity typed as ASCII.

    s"   The symbol s, of one or two characters right-adjusted, is taken as a term on input.

- To print a quantity in the current master print mode:

    ;   Print last quantity typed in current mode.

- To change the breakpoint register print mode to current master print mode:

    \O

- To operate in character or block mode (multipoint terminals). Refer to Appendix D.

    \\U   Switch from character to block mode or vice versa.

- To change a print mode temporarily:

$ Same as bar (/), but set termporary print mode to symbolic instruction. Temporary mode is in effect over a series of examinations, until carriage return is typed by you, then the master mode becomes in effect again.

\$ Same as escape bar, but set temporary print mode to symbolic instruction.

@   Same as bar, but set temporary print mode to address pointer.

\@   Same as escape bar, but set temporary print mode to address pointer.

‡   Same as bar, but set temporary print mode to constant.

\‡   Same as escape bar, but set temporary print mode to constant.

)   Same as bar, but set temporary print mode to ASCII.

\)   Same as escape bar, but set temporary print mode to ASCII.

C-2. SYMBOL MANIPULATION

s: Define the symbol s as having value specified by the location counter.

s&   Define symbol s as having value equal to the address of the last quantity typed, if an instruction.

\K   Kill all symbol definitions other than the initial symbol table.

C-3. LOCATION EXAMINATION

n/   Print the contents of location n, in the current master print mode, and open the location for possible modification. The location counter is set to n.

/   Open and print the contents of the location pointed to by the last quantity typed, taken as a direct 15-bit address. The location counter is not changed.

\/   Open and print the contents of the location pointed to by the last quantity typed, taken as an memory reference address instruction. The location counter is not changed.

\L   Open and print the contents of the next 16 sequential locations starting with the current location. The location counter is advanced by 16.

n\L   Same as \L, except list n lines.

[CR]   Close any open location. No change is made.

n[CR]   Store the quantity n in the open location, if any.

[LF]   Open and print the contents of the next sequential location as determined by the location counter. The location counter is advanced by one.

n[LF]   Store the quantity n in the open location, if any; then open and print the contents of the next sequential location. The location counter is advanced by one.

^   Open and print the contents of the previous sequential register. The location counter is decremented by one.

n^   Store the quantity n in the open location, if any; then open and print the contents of the previous sequential location. The location counter is decremented by one.

CTRL-I   Open, set the location counter to, and print the contents
(TAB)   of the location pointed to by the quantity typed, taken as an address pointer.

nCTRL-I   Store the quantity n in the open location, if any. Then
(nTAB)   open, set the loction counter to, and print the contents of the location pointed to by n, taken as an address.

\TAB   Open, set the location counter to, and print the contents of the location pointed to by the last quantity typed, taken as an instruction.

n\TAB   Store the quantity n in the open location, if any. Then open, set the location counter to, and print the contents of the location pointed by n, taken as an instruction.

n%   Store the quantity n in the open location, if any. Then open and print the contents of the location pointed to by n, taken as an address. The location counter is not changed.

n\%   Store the quantity n in the open location, if any. Then open and print the contents of the location pointed to by n, taken as an instruction. The location counter is not changed.

C-4. PROGRAM LOAD AND VERIFY

\Y   Load absolute binary paper tape, as prepared by HP
              Assembler.

\V   Verify paper tape against core.

C-5. PUNCHING

n\D   Store n into the open location, if any. Then punch the
              last register (with the new contents n, if stored).

n1<n2>\D   Punch locations n1 to n2, inclusive.

C-6. MEMORY SEARCH AND CLEAR

n<   Set the lower limit for search or clear to n.

n>   Set the upper limit for search or clear to n.

n1<n2>n3\W   Search between the limits n1 and n2 for all words equal
              under mask to n3.

n1<n2>n3\N   Search between the limits n1 and n2 for all words not
              equal under mask to n3.

n1<n2>n3\E   Effective   address   search: find all   instructions
              between the limits n1 and n2 which, under mask,
              effectively address location n3.

n1<n2>n3\\Z   Zero, or clear core between the limits n1 and n2, to
              n3. If n3 is omitted, it is taken as zero.

\MASK   Examine/modify search mask.

C-7. BREAKPOINTS AND PROGRAM CONTROL

n\B   Set a breakpoint at n. If n is in segment space, use
              last segment loaded as the associated segment name.
              Error if no segment loaded yet.

\B   Enter remove breakpoint mode.

\\B   Remove all breakpoints

n[NAME]\B   Set a breakpoint in segment NAME at location n.

[NAME]\B   Break at entry to Segment NAME.

["A]\B   Break at entry to ALL Segments.

["N]\B	Break at entry to NO Segments except for those specified.
\P	Proceed with program execution after a break trap.
\\P	Proceed with conditional breakpoint invoked.
n\P	Proceed; do not trap until n breakpoints from now.
n\\P	Proceed; do not trap until n breakpoints, including conditional breakpoints.
n\G	Go to location n; begin execution with flags and accumulators as saved.
n\X	Execute the instruction n, then return control to DBUGR.
\T	Trace one instruction.
n\T	Trace n instructions.
\\T	Trace an entire subroutine call with no argument list or alternate returns.

Trace/Breakpoint simulation:

DBUGR simulates instructions when it:

a)   Proceeds from the current breakpoint location

b)   Traces

c)   Executes an instruction

DBUGR will correctly simulate all instructions including EXEC calls in its symbol table when tracing, with the exceptions noted in Figure 6-1. A breakpoint may not be set on any of these instructions.

A method to get around instruction exceptions if to set up a subroutine containing the desired instruction(s) and execute a JSB <subroutine>\X.   DBUGR will give up control to the subroutine and execute it, regaining control on the return.

C-8. SPECIAL REGISTERS

```
\M        Display contents of the special registers.

AREG/    Examine & Modify A-Register

BREG/    Examine & Modify B-Register

XREG/    Examine & Modify X-Register

YREG/    Examine & Modify Y-Register

EOREG/   Examine & Modify EO-Register

MASK/    Examine & Modify Search Mask

CBVAL/   Examine & Modify conditional breakpoint value

CBMASK/ Examine & Modify conditional breakpoint mask

CBADDR/ Examine & Modify conditional breakpoint address

CBTEST/ Examine & Modify conditional breakpoint test

WRTLU/   EXEC control word for DBUGR output device

BRFLG/  Break Flag; 0= check for break, 1= no check
```

C-9. MAP REGISTERS

```
\J:   SM  Examine system maps.

      UM  Examine user maps.

      PA  Examine port A maps.

      PB  Examine port B maps.

      XL  Cross load A- or B-registers.

\?   Display MEM status register.
```

DBUGR can be operated in block mode. Multipoint (DVR07) terminals run in block mode only.

Escape, return and line feed cannot be used in block mode.

The following are the differences between character and block mode.

```
    CHAR MODE              BLOCK MODE
    ---------              ----------
    <escape>                  \
      <cr>                    ]
      <lf>                 <enter for multipoint>
                           <return for non-multipoint>
```

In order to invoke block mode operation on a non-multipoint terminal enter the following command:

   \\U

When in block mode this command will set DBUGR back to character mode for non-multipoint terminals.

Since DBUGR does not receive any character until the enter or return is entered, a series of commands may be entered on one line.

For example:

```
    200/201/202/<enter>
    -------------------
        JSB 4,I   203   245
    200/<enter>
    -----------
        JSB 4,I   <enter>
                  -------
    201/ 203      <enter>
                  -------
    202/ 245      ]<enter>
                  -------
```

READER COMMENT SHEET

RTE-IV DEBUG SUBROUTINE
Reference Manual

92067-90005   **February 1980**

Update No. _____

(If Applicable)

We welcome your evaluation of this manual. Your comments and suggestions help us improve our publications. Please use additional pages if necessary.

_____

FROM:

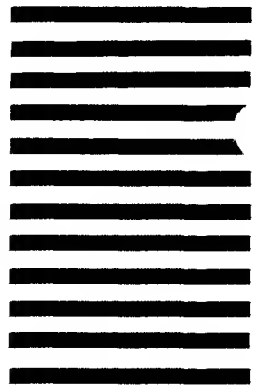**Name**   _____

Company   _____

Address   _____

                _____

                _____

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 141 CUPERTINO, CA.

— POSTAGE WILL BE PAID BY —

Hewlett-Packard Company
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014
ATTN: Technical Marketing Dept.

Product Line Sales/Support Key

Key Product Line
A   Analytical
CM   Components
C   Computer Systems Sales only
CH   Computer Systems Hardware Sales and Services
CS   Computer Systems Software Sales and Services
E   Electronic Instruments & Measurement Systems
M   Medical Products
MP   Medical Products Primary SRO
MS   Medical Products Secondary SRO
P   Personal Computation Products
·   Sales only for specific product line
··   Support only for specific product line

IMPORTANT: These symbols designate general product line capability. They do not insure sales or support availability for all products within a line, at all locations. Contact your local sales office for information regarding locations where HP support is available for specific products.

HP distributors are printed in italics.

HEADQUARTERS OFFICES
If there is no sales office listed for your area, contact one of these headquarters offices.

NORTH/CENTRAL AFRICA
Hewlett-Packard S.A.
7, Rue du Bois-du-Lan
CH-1217 MEYRIN 2, Switzerland
Tel: (022) 83 12 12
Telex: 27835 hpse
Cable: HEWPACKSA Geneve

ASIA
Hewlett-Packard Asia Ltd.
6th Floor, Sun Hung Kai Centre
30 Harbour Rd.
G.P.O. Box 795
HONG KONG
Tel: 5-832 3211
After Jan. 1, 1984
47th Floor, China Resources Bldg.
26 Harbour Rd., Wanchai
HONG KONG
Telex: 66678 HEWPA HX
Cable: HEWPACK HONG KONG

CANADA
Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
MISSISSAUGA, Ontario L4V 1M8
Tel: (416) 678-9430
Telex: 610-492-4246

EASTERN EUROPE
Hewlett-Packard Ges.m.b.H.
Lieblgasse 1
P.O.Box 72
A-1222 VIENNA, Austria
Tel: (222) 2365110
Telex: 1 3 4425 HEPA A

NORTHERN EUROPE
Hewlett-Packard S.A.
Uilenstede 475
P.O.Box 999
NL-1180 AZ AMSTELVEEN
The Netherlands
Tel: 20 437771

SOUTH EAST EUROPE
Hewlett-Packard S.A.
7, Rue du Bois-du-Lan
CH-1217 MEYRIN 2, Switzerland
Tel: (022) 83 12 12
Telex: 27835 hpse
Cable: HEWPACKSA Geneve

OTHER EUROPE
Hewlett-Packard S.A.
P.O. Box
150, Rte du Nant-O'Avril
CH-1217 MEYRIN 2, Switzerland
Tel: (022) 83 8111
Telex: 22486 hpsa
Cable: HEWPACKSA Geneve

MEDITERRANEAN AND MIDDLE EAST
Hewlett-Packard S.A.
Mediterranean and Middle East
Operations
Atrina Centre
32 Kifissias Ave.
Paradissos-Amarousion, ATHENS
Greece
Tel: 682 88 11
Telex: 21-6588 HPAT GR
Cable: HEWPACKSA Athens

EASTERN USA
Hewlett-Packard Co.
4 Choke Cherry Road
ROCKVILLE, MD 20850
Tel: (301) 258-2000

MIDWESTERN USA
Hewlett-Packard Co.
5201 Tollview Drive
ROLLING MEADOWS, IL 60008
Tel: (312) 255-9800

SOUTHERN USA
Hewlett-Packard Co.
2000 South Park Place
P.O. Box 105005
ATLANTA, GA 30348
Tel: (404) 955-1500

WESTERN USA
Hewlett-Packard Co.
3939 Lankershim Blvd.
P.O. Box 3919
LOS ANGELES, CA 91604
Tel: (213) 506-3700

OTHER INTERNATIONAL AREAS
Hewlett-Packard Co.
Intercontinental Headquarters
3495 Deer Creek Road
PALO ALTO, CA 94304
Tel: (415) 857-1501
Telex: 034-8300
Cable: HEWPACK

ANGOLA
Telectra
Empresa Técnica de Equipamentos
R. Barbosa Rodrigues, 41-I DT.
Caixa Postal 6487
LUANDA
Tel: 35515,35516
E,P

ARGENTINA
Hewlett-Packard Argentina S.A.
Avenida Santa Fe 2035
Martinez 1640 BUENOS AIRES
Tel: 798-5735, 792-1293
Telex: 17595 BIONAR
Cable: HEWPACKARG
A,E,CH,CS,P

Biotron S.A.C.I.M. e I.
Av Paseo Colon 221, Piso 9
1399 BUENOS AIRES
Tel: 30-4846, 30-1851
Telex: 17595 BIONAR
M

AUSTRALIA

Adelaide, South Australia Office
Hewlett-Packard Australia Ltd.
153 Greenhill Road
PARKSIDE, S.A. 5063
Tel: 272-5911
Telex: 82536
Cable: HEWPARO Adelaide
A·,CH,CM,,E,MS,P

Brisbane, Queensland Office
Hewlett-Packard Australia Ltd.
10 Payne Road
THE GAP, Queensland 4061
Tel: 30-4133
Telex: 42133
Cable: HEWPARO Brisbane
A,CH,CM,E,M,P

Canberra, Australia Capital Territory Office
Hewlett-Packard Australia Ltd.
121 Wollongong Street
FYSHWICK, A.C.T. 2609
Tel: 80 4244
Telex: 62650
Cable: HEWPARO Canberra
CH,CM,E,P

Melbourne, Victoria Office
Hewlett-Packard Australia Ltd.
31-41 Joseph Street
BLACKBURN, Victoria 3130
Tel: 895-2895
Telex: 31-024
Cable: HEWPARO Melbourne
A,CH,CM,CS,E,MS,P

Perth, Western Australia Office
Hewlett-Packard Australia Ltd.
261 Stirling Highway
CLAREMONT, W.A. 6010
Tel: 383-2188
Telex: 93859
Cable: HEWPARO Perth
A,CH,CM,E,MS,P

Sydney, New South Wales Office
Hewlett-Packard Australia Ltd.
17-23 Talavera Road
P.O. Box 308
NORTH RYDE, N.S.W. 2113
Tel: 887-1611
Telex: 21561
Cable: HEWPARO Sydney
A,CH,CM,CS,E,MS,P

AUSTRIA
Hewlett-Packard Ges.m.b.h.
Grottenhofstrasse 94
A-8052 GRAZ
Tel: (0316) 291 5 66
Telex: 32375
CH,E

Hewlett-Packard Ges.m.b.h.
Lieblgasse 1
P.O. Box 72
A-1222 VIENNA
Tel: (0222) 23 65 11-0
Telex: 134425 HEPA A
A,CH,CM,CS,E,MS,P

BAHRAIN
Green Salon
P.O. Box 557
Manama
BAHRAIN
Tel: 255503-255950
Telex: 84419
P

Wael Pharmacy
P.O. Box 648
BAHRAIN
Tel: 256123
Telex: 8550 WAEL BN
E,C,M

BELGIUM
Hewlett-Packard Belgium S.A./N.V.
Blvd de la Woluwe, 100
Woluwedal
B-1200 BRUSSELS
Tel: (02) 762-32-00
Telex: 23-494 paloben bru
A,CH,CM,CS,E,MP,P

BRAZIL
Hewlett-Packard do Brasil I.e.C. Ltda.
Alameda Rio Negro, 750
Alphaville
06400 BARUERI SP
Tel: (011) 421.1311
Telex: (011) 33872 HPBR-BR
Cable: HEWPACK Sao Paulo
A,CH,CM,CS,E,M,P

Hewlett-Packard do Brasil I.e.C. Ltda.
Avenida Epitacio Pessoa, 4664
22471 RIO DE JANEIRO-RJ
Tel: (02I) 286.0237
Telex: 021-21905 HPBR-BR
Cable: HEWPACK Rio de Janeiro
A,CH,CM,E,MS,P·

ANAMED I.C.E.I. Ltda.
Rua Bage, 103
04012 SAO PAULO
Tel: (011) 570-5726
Telex: 021-21905 HPBR-BR
M

SALES & SUPPORT OFFICES
Arranged alphabetically by country

CANADA

Alberta
Hewlett-Packard (Canada) Ltd.
3030 3rd Avenue N.E.
CALGARY, Alberta T2A 6T7
Tel: (403) 235-3100
A,CH,CM,E*,MS,P*

Hewlett-Packard (Canada) Ltd.
11120A-178th Street
EDMONTON, Alberta T5S 1P2
Tel: (403) 486-6666
A,CH,CM,CS,E,MS,P

British Columbia
Hewlett-Packard (Canada) Ltd.
10691 Shellbridge Way
RICHMOND,
British Columbia V6X 2W7
Tel: (604) 270-2277
Telex: 610-922-5059
A,CH,CM,CS,E*,MS,P*

Manitoba
Hewlett-Packard (Canada) Ltd.
380-550 Century Street
WINNIPEG, Manitoba R3H 0Y1
Tel: (204) 786-6701
A,CH,CM,E,MS,P*

Nova Scotia
Hewlett-Packard (Canada) Ltd.
P.O. Box 931
900 Windmill Road
DARTMOUTH, Nova Scotia B2Y 3Z6
Tel: (902) 469-7820
CH,CM,CS,E*,MS,P*

Ontario
Hewlett-Packard (Canada) Ltd.
3325 N. Service Rd., Unit 6
BURLINGTON, Ontario P3A 2A3
Tel: (416) 335-8644
CS,M*

Hewlett-Packard (Canada) Ltd.
552 Newbold Street
LONDON, Ontario N6E 2S5
Tel: (519) 686-9181
A,CH,CM,E*,MS,P*

Hewlett-Packard (Canada) Ltd.
6877 Goreway Orive
MISSISSAUGA, Ontario L4V 1M8
Tel: (416) 678-9430
A,CH,CM,CS,E,MP,P

Hewlett-Packard (Canada) Ltd.
2670 Queensview Or.
OTTAWA, Ontario K2B 8K1
Tel: (613) 820-6483
A,CH,CM,CS,E*,MS,P*

Hewlett-Packard (Canada) Ltd.
220 Yorkland Blvd., Unit #11
WILLOWDALE, Ontario M2J 1R5
Tel: (416) 499-9333
CH

Quebec
Hewlett-Packard (Canada) Ltd.
17500 South Service Road
Trans-Canada Highway
KIRKLAND, Quebec H9J 2M5
Tel: (514) 697-4232
A,CH,CM,CS,E,MP,P*

Hewlett-Packard (Canada) Ltd.
Les Galeries du Vallon
2323 Ou Versont Nord
STE. FOY, Quebec G1N 4C2
Tel: (418) 687-4570
CH

CHILE
Jorge Calcagni y Cia. Ltda.
Av. Italia 634 Santiago
Casilla 16475
SANTIAGO 9
Tel: 222-0222
Telex: Public Booth 440001
A,CM,E,M

Olympia (Chile) Ltda.
Av. Rodrigo de Araya 1045
Casilla 256-V
SANTIAGO 21
Tel: (02) 22 55 044
Telex: 240-565 OLYMP CL
Cable: Olympiachile Santiagochile
CH,CS,P

CHINA, People's Republic of
China Hewlett-Packard Rep. Office
P.O. Box 418
1A Lane 2, Luchang St.
Beiwei Rd., Xuanwu District
BEIJING
Tel: 33-1947, 33-7426
Telex: 22601 CTSHP CN
Cable: 1920
A,CH,CM,CS,E,P

COLOMBIA
Instrumentación
H. A. Langebaek & Kier S.A.
Carrera 4A No. 52A-26
Apartado Aereo 6287
BOGOTA 1, D.E.
Tel: 212-1466
Telex: 44400 INST CO
Cable: AARIS Bogota
CM,E,M

Casa Humboldt Ltda.
Carrera 14, No. 98-60
Apartado Aereo 51283
BOGOTA 1, D.E.
Tel: 256-1686
Telex: 45403 CCAL CO.
A

COSTA RICA
Cientifica Costarricense S.A.
Avenida 2, Calle 5
San Pedro de Montes de Oca
Apartado 10159
SAN JOSÉ
Tel: 24-38-20, 24-08-19
Telex: 2367 GALGUR CR
CM,E,M

CYPRUS
Telerexa Ltd.
P.O. Box 4809
14C Stassinos Avenue
NICOSIA
Tel: 62698
Telex: 2894 LEVIDO CY
E,M,P

DENMARK
Hewlett-Packard A/S
Datavej 52
DK-3460 **BIRKEROD**
Tel: (02) 81-66-40
Telex: 37409 hpas dk
A,CH,CM,CS,E,MS,P

Hewlett-Packard A/S
Rolighedsvej 32
OK-8240 **RISSKOV**, Aarhus
Tel: (06) 17-60-00
Telex: 37409 hpas dk
CH,E

DOMINICAN REPUBLIC
Microprog S.A.
Juan Tomás Mejía y Cotes No. 60
Arroyo Hondo
SANTO DOMINGO
Tel: 565-6268
Telex: 4510 ARENTA DR (RCA) P

ECUADOR
CYEDE Cia. Ltda.
Avenida Eloy Alfaro 1749
Casilla 6423 CCI
QUITO
Tel: 450-975, 243-052
Telex: 2548 CYEDE ED
CM,E,P

Hospitalar S.A.
Robles 625
Casilla 3590
QUITO
Tel: 545-250, 545-122
Telex: 2485 HOSPTL ED
Cable: HOSPITALAR-Quito
M

EGYPT
International Engineering Associates
24 Hussein Hegazi Street
Kasr-el-Aini
CAIRO
Tel: 23829, 21641
Telex: IEA UN 93830
CH,CS,E,M

EGYPOR
P.O.Box 2558
42 El Zahraa Street
CAIRO, Egypt
Tel: 65 00 21
Telex: 93 337
P

EL SALVADOR
IPESA de El Salvador S.A.
29 Avenida Norte 1216
SAN SALVADOR
Tel: 26-6858, 26-6868
Telex: 20539 IPESASAL
A,CH,CM,CS,E,P

FINLAND
Hewlett-Packard Oy
Revontulentie 7
PL 24
SF-02101 **ESPOO** 10
Tel: (90) 4550211
Telex: 121563 hewpa sf
CH,CM,CS,P

Hewlett-Packard Oy
(Olariniuoma 7)
PL 24
02101 **ESPOO** 10
Tel: (90) 4521022
A,E,MS

Hewlett-Packard Oy
Aatoksenkatv 10-C
SF-40720-72 **JYVASKYLA**
Tel: (941) 216318
CH

Hewlett-Packard Oy
Kainvuntie 1-C
SF-90140-14 **OULU**
Tel: (981) 338785
CH

FRANCE
Hewlett-Packard France
Z.I. Mercure B
Rue Berthelot
F-13763 Les Milles Cedex
AIX-EN-PROVENCE
Tel: 16 (42) 59-41-02
Telex: 410770F
A,CH,E,MS,P*

Hewlett-Packard France
64, rue Marchand Saillant
F-61000 **ALENCON**
Tel: 16 (33) 29 04 42

Hewlett-Packard France
Boite Postale 503
F-25026 **BESANCON**
28 rue de la Republique
F-25000 **BESANCON**
Tel: 16 (81) 83-16-22
CH,M

Hewlett-Packard France
13, Place Napoleon III
F-29000 **BREST**
Tel: 16 (98) 03-38-35

Hewlett-Packard France
Chemin des Mouilles
Boite Postale 162
F-69130 **ECULLY** Cedex (Lyon)
Tel: 16 (78) 833-81-25
Telex: 310617F
A,CH,CS,E,MP

Hewlett-Packard France
Tour Lorraine
Boulevard de France
F-91035 **EVRY** Cedex
Tel: 16 6 077-96-60
Telex: 692315F
E

Hewlett-Packard France
Parc d'Activite du Bois Briard
Ave. du Lac
F-91040 **EVRY** Cedex
Tel: 16 6 077-8383
Telex: 692315F
E

Hewlett-Packard France
5, avenue Raymond Chanas
F-38320 **EYBENS** (Grenoble)
Tel: 16 (76) 25-81-41
Telex: 980124 HP GRENOB EYBE
CH

Hewlett-Packard France
Centre d'Affaire Paris-Nord
Bâtiment Ampère 5 étage
Rue de la Commune de Paris
Boite Postale 300
F-93153 **LE BLANC MESNIL**
Tel: 16 (1) 865-44-52
Telex: 211032F
CH,CS,E,MS

Hewlett-Packard France
Parc d'Activités Cadera
Quartier Jean Mermoz
Avenue du Président JF Kennedy
F-33700 **MERIGNAC** (Bordeaux)
Tel: 16 (56) 34-00-84
Telex: 550105F
CH,E,MS

Hewlett-Packard France
Immeuble "Les 3 B"
Nouveau Chemin de la Garde
ZAC de Bois Briand
F-44085 **NANTES** Cedex
Tel: 16 (40) 50-32-22
CH**

FRANCE (Cont'd)
Hewlett-Packard France
125, rue du Faubourg Bannier
F-45000 ORLEANS
Tel: 16 (38) 68 01 63
Hewlett-Packard France
Zone Industrielle de Courtaboeuf
Avenue des Tropiques
F-91947 Les Ulis Cedex ORSAY
Tel: (6) 907-78-25
Telex: 600048F
A,CH,CM,CS,E,MP,P
Hewlett-Packard France
Paris Porte-Maillot
15, Avenue de L'Amiral Bruix
F-75782 PARIS CEDEX 16
Tel: 16 (1) 502-12-20
Telex: 613663F
CH,MS,P
Hewlett-Packard France
124, Boulevard Tourasse
F-64000 PAU
Tel: 16 (59) 80 38 02
Hewlett-Packard France
2 Allée de la Bourgonnette
F-35100 RENNES
Tel: 16 (99) 51-42-44
Telex: 740912F
CH,CM,E,MS,P*
Hewlett-Packard France
98 Avenue de Bretagne
F-76100 ROUEN
Tel: 16 (35) 63-57-66
CH**,CS
Hewlett-Packard France
4 Rue Thomas Mann
Boite Postale 56
F-67033 STRASBOURG Cedex
Tel: 16 (88) 28-56-46
Telex: 890141F
CH,E,MS,P*
Hewlett-Packard France
Le Péripole
20, Chemin du Pigeonnier de la Cépière
F-31083 TOULOUSE Cedex
Tel: 16 (61) 40-11-12
Telex: 531639F
A,CH,CS,E,P*
Hewlett-Packard France
9, rue Baudin
F-26000 VALENCE
Tel: 16 (75) 42 76 16
Hewlett-Packard France
Carolor
ZAC de Bois Briand
F-57640 VIGY (Metz)
Tel: 16 (8) 771 20 22
CH
Hewlett-Packard France
Immeuble Péricentre
F-59658 VILLENEUVE D'ASCQ Cedex
Tel: 16 (20) 91-41-25
Telex: 160124F
CH,E,MS,P*

GERMAN FEDERAL REPUBLIC
Hewlett-Packard GmbH
Geschäftsstelle
Keithstrasse 2-4
D-1000 BERLIN 30
Tel: (030) 24-90-86
Telex: 018 3405 hpbln d
A,CH,E,M,P

Hewlett-Packard GmbH
Geschäftsstelle
Herrenberger Strasse 130
D-7030 BÖBLINGEN
Tel: (7031) 14-0
Telex:
A,CH,CM,CS,E,MP,P
Hewlett-Packard GmbH
Geschäftsstelle
Emanuel-Leutze-Strasse 1
D-4000 DUSSELDORF
Tel: (0211) 5971-1
Telex: 085/86 533 hpdd d
A,CH,CS,E,MS,P
Hewlett-Packard GmbH
Geschäftsstelle
Schleefstr. 28a
D-4600 DORTMUND-Aplerbeck
Tel: (0231) 45001
Hewlett-Packard GmbH
Vertriebszentrale Frankfurt
Berner Strasse 117
Postfach 560 140
D-6000 FRANKFURT 56
Tel: (0611) 50-04-1
Telex: 04 13249 hpffm d
A,CH,CM,CS,E,MP,P
Hewlett-Packard GmbH
Geschäftsstelle
Aussenstelle Bad Homburg
Louisenstrasse 115
D-6380 BAD HOMBURG
Tel: (06172) 109-0
Hewlett-Packard GmbH
Geschäftsstelle
Kapstadtring 5
D-2000 HAMBURG 60
Tel: (040) 63804-1
Telex: 021 63 032 hphh d
A,CH,CS,E,MS,P
Hewlett-Packard GmbH
Geschäftsstelle
Heidering 37-39
D-3000 HANNOVER 61
Tel: (0511) 5706-0
Telex: 092 3259
A,CH,CM,E,MS,P
Hewlett-Packard GmbH
Geschäftsstelle
Rosslauer Weg 2-4
D-6800 MANNHEIM
Tel: (0621) 70050
Telex: 0462105
A,C,E
Hewlett-Packard GmbH
Geschäftsstelle
Messerschmittstrasse 7
D-7910 NEU ULM
Tel: 0731-70241
Telex: 0712816 HP ULM-D
A,C,E*
Hewlett-Packard GmbH
Geschäftsstelle
Ehhericherstr. 13
D-8500 NÜRNBERG 10
Tel: (0911) 5205-0
Telex: 0623 860
CH,CM,E,MS,P
Hewlett-Packard GmbH
Geschäftsstelle
Eschenstrasse 5
D-8028 TAUFKIRCHEN
Tel: (089) 6117-1
Telex: 0524985
A,CH,CM,E,MS,P

GREAT BRITAIN
See United Kingdom

GREECE
Kostas Karaynnis S.A.
8 Omirou Street
ATHENS 133
Tel: 32 30 303, 32 37 371
Telex: 215962 RKAR GR
A,CH,CM,CS,E,M,P
PLAISIO S.A.
G. Gerardos
24 Stournara Street
ATHENS
Tel: 36-11-160
Telex: 221871
P

GUATEMALA
IPESA
Avenida Reforma 3-48, Zona 9
GUATEMALA CITY
Tel: 316627, 314786
Telex: 4192 TELTRO GU
A,CH,CM,CS,E,M,P

HONG KONG
Hewlett-Packard Hong Kong, Ltd.
G.P.O. Box 795
5th Floor, Sun Hung Kai Centre
30 Harbour Road
HONG KONG
Tel: 5-8323211
Telex: 66678 HEWPA HX
Cable: HEWPACK HONG KONG
E,CH,CS,P
CET Ltd.
1402 Tung Wah Mansion
199-203 Hennessy Rd.
Wanchia, HONG KONG
Tel: 5-729376
Telex: 85148 CET HX
CM
Schmidt & Co. (Hong Kong) Ltd.
Wing On Centre, 28th Floor
Connaught Road, C.
HONG KONG
Tel: 5-455644
Telex: 74766 SCHMX HX
A,M

ICELAND
Elding Trading Company Inc.
Hafnarnvoli-Tryggvagotu
P.O. Box 895
IS-REYKJAVIK
Tel: 1-58-20, 1-63-03
M

INDIA
Computer products are sold through Blue Star Ltd. All computer repairs and maintenance service is done through Computer Maintenance Corp.
Blue Star Ltd.
Sabri Complex II Floor
24 Residency Rd.
BANGALORE 560 025
Tel: 55660
Telex: 0845-430
Cable: BLUESTAR
A,CH,CM,CS*,E*

Blue Star Ltd.
Band Box House
Prabhadevi
BOMBAY 400 025
Tel: 422-3101
Telex: 011-3751
Cable: BLUESTAR
A,M
Blue Star Ltd.
Sahas
414/2 Vir Savarkar Marg
Prabhadevi
BOMBAY 400 025
Tel: 422-6155
Telex: 011-4093
Cable: FROSTBLUE
A,CH,CM,CS*,E,M*
Blue Star Ltd.
Kalyan, 19 Vishwas Colony
Alkapuri, BORODA, 390 005
Tel: 65235
Cable: BLUE STAR
A
Blue Star Ltd.
7 Hare Street
CALCUTTA 700 001
Tel: 12-01-31
Telex: 021-7655
Cable: BLUESTAR
A,M
Blue Star Ltd.
133 Kodambakkam High Road
MADRAS 600 034
Tel: 82057
Telex: 041-379
Cable: BLUESTAR
A,M
Blue Star Ltd.
Bhandari House, 7th/8th Floors
91 Nehru Place
NEW DELHI 110 024
Tel: 682547
Telex: 031-2463
Cable: BLUESTAR
A,CH,CM,CS*,E,M*
Blue Star Ltd.
15/16:C Wellesley Rd.
PUNE 411 011
Tel: 22775
Cable: BLUE STAR
A
Blue Star Ltd.
2-2-47/1108 Bolarum Rd.
SECUNDERABAD 500 003
Tel: 72057
Telex: 0155-459
Cable: BLUEFROST
A,E
Blue Star Ltd.
T.C. 7/603 Poornima
Maruthankuzhi
TRIVANDRUM 695 013
Tel: 65799
Telex: 0884-259
Cable: BLUESTAR
E
Computer Maintenance Corporation Ltd.
115, Sarojini Devi Road
SECUNDERABAD 500 003
Tel: 310-184, 345-774
Telex: 031-2960
*CH**

INDONESIA
BERCA Indonesia P.T.
P.O.Box 496/Jkt.
Jl. Abdul Muis 62
JAKARTA
Tel: 21-373009
Telex: 46748 BERSAL IA
Cable: BERSAL JAKARTA
P

BERCA Indonesia P.T.
P.O.Box 2497/Jkl
Antara Bldg., 17th Floor
Jl. Medan Merdeka Selalan 17
JAKARTA-PUSAT
Tel: 21-344-181
Telex: BERSAL IA
A,CS,E,M

BERCA Indonesia P.T.
P.O. Box 174/SBY.
Jl. Kulei No. 11
SURABAYA
Tel: 68172
Telex: 31146 BERSAL SB
Cable: BERSAL-SURABAYA
A*,E,M,P

IRAQ
Hewlett-Packard Trading S.A.
Service Operation
Al Mansoor City 9B/3/7
BAGHDAD
Tel: 551-49-73
Telex: 212-455 HEPAIRAQ IK
CH,CS

IRELAND
Hewlett-Packard Ireland Ltd.
82/83 Lower Leeson Street
DUBLIN 2
Tel: 0001 608800
Telex: 30439
A,CH,CM,CS,E,M,P

Cardiac Services Ltd.
Kilmore Road
Artane
DUBLIN 5
Tel: (01) 351820
Telex: 30439
M

ISRAEL
Eldan Electronic Inslrumenl Ltd.
P.O.Box 1270
JERUSALEM 91000
16, Ohaliav St.
JERUSALEM 94467
Tel: 533 221, 553 242
Telex: 25231 AB/PAKRD IL
A

Electronics Engineering Division
Motorola Israel Ltd.
16 Kremenetski Slreet
P.O. Box 25016
TEL-AVIV 67899
Tel: 3 88 388
Telex: 33569 Molil IL
Cable: BASTEL Tel-Aviv
CH,CM,CS,E,M,P

ITALY
Hewlett-Packard Italiana S.p.A
Traversa 99C
Via Giulio Petroni, 19
I-70124 **BARI**
Tel: (080) 41-07-44
M

Hewlett-Packard Italiana S.p.A.
Via Martin Luther King, 38/III
I-40132 **BOLOGNA**
Tel: (051) 402394
Telex: 511630
CH,E,MS

Hewlett-Packard Italiana S.p.A.
Via Principe Nicola 43G/C
I-95126 **CATANIA**
Tel: (095) 37-10-87
Telex: 970291
C,P

Hewlett-Packard Italiana S.p.A.
Via G. Oi Vittorio 9
I-20063 **CERNUSCO SUL NAVIGLIO**
(Milano)
Tel: (02) 923691
Telex: 334632
A,CH,CM,CS,E,MP,P

Hewlett-Packard ttaliana S.p.A.
Via C. Colombo 49
I-20090 **TREZZANO SUL NAVIGLIO**
(Milano)
Tel: (02) 4459041
Telex: 322116
C,M

Hewlett-Packard Italiana S.p.A.
Via Nuova San Rocco a
Capodimonte, 62/A
I-80131 **NAPOLI**
Tel: (081) 7413544
Telex: 710698
A,CH,E

Hewlett-Packard Italiana S.p.A.
Viale G. Modugno 33
I-16156 **GENOVA PEGLI**
Tel: (010) 68-37-07
Telex: 215238
E,C

Hewlett-Packard Italiana S.p.A.
Via Pelizzo 15
I-35128 **PADOVA**
Tel: (049) 664888
Telex: 430315
A,CH,E,MS

Hewlett-Packard Italiana S.p.A.
Viale C. Pavese 340
I-00144 **ROMA EUR**
Tel: (06) 54831
Telex: 610514
A,CH,CM,CS,E,MS,P*

Hewlett-Packard Italiana S.p.A.
Via di Casellina 57/C
I-50018 **SCANDICCI-FIRENZE**
Tel: (055) 753863

Hewlett-Packard Italiana S.p.A.
Corso Svizzera, 185
I-10144 **TORINO**
Tel: (011) 74 4044
Telex: 221079
CH,E

JAPAN
Yokogawa-Hewlett-Packard Ltd.
152-1, Onna
ATSUGI, Kanagawa, 243
Tel: (0462) 28-0451
CM,C*,E

Yokogawa-Helwett-Packard Ltd.
Meiji-Seimei Bldg. 6F
3-1 Hon Chiba-Cho
CHIBA, 280
Tel: 472 25 7701
E,CH,CS

Yokogawa-Hewlett-Packard Ltd.
Yasuda-Seimei Hiroshima Bldg.
6-11, Hon-dori, Naka-ku
HIROSHIMA, 730
Tel: 82-241-0611

Yokogawa-Hewlett-Packard Ltd.
Towa Building
2-3, Kaigan-dori, 2 Chome Chuo-ku
KOBE, 650
Tel: (078) 392-4791
C,E

Yokogawa-Hewlett-Packard Ltd.
Kumagaya Asahi 82 Bldg
3-4 Tsukuba
KUMAGAYA, Saitama 360
Tel: (0485) 24-6563
CH,CM,E

Yokogawa-Hewlett-Packard Ltd.
Asahi Shinbun Oaiichi Seimei Bldg.
4-7, Hanabata-cho
KUMAMOTO,860
Tel: (0963) 54-7311
CH,E

Yokogawa-Hewlett-Packard Ltd.
Shin-Kyoto Center Bldg.
614, Higashi-Shiokoji-cho
Karasuma-Nishiiru
Shiokoji-dori, Shimogyo-ku
KYOTO, 600
Tel: 075-343-0921
CH,E

Yokogawa-Hewlett-Packard Ltd.
Mito Mitsui Bldg
4-73, Sanno-maru, 1 Chome
MITO, Ibaraki 310
Tel: (0292) 25-7470
CH,CM,E

Yokogawa-Hewlett-Packard Ltd.
Sumitomo Seimei 14-9 Bldg.
Meieki-Minami, 2 Chome
Nakamura-ku
NAGOYA, 450
Tel: (052) 571-5171
CH,CM,CS,E,MS

Yokogawa-Hewlett-Packard Ltd.
Chuo Bldg.,
4-20 Nishinakajima, 5 Chome
Yodogawa-ku
OSAKA, 532
Tel: (06) 304-6021
Telex: YHPOSA 523-3624
A,CH,CM,CS,E,MP,P*

Yokogawa-Hewlett-Packard Ltd.
27-15, Yabe, 1 Chome
SAGAMIHARA Kanagawa, 229
Tel: 0427 59-1311

Yokogawa-Hewlett-Packard Ltd.
Oaiichi Seimei Bldg.
7-1, Nishi Shinjuku, 2 Chome
Shinjuku-ku,**TOKYO** 160
Tel: 03-348-4611
CH,E

Yokogawa-Hewlett-Packard Ltd.
29-21 Takaido-Higashi, 3 Chome
Suginami-ku **TOKYO** 168
Tel: (03) 331-611
Telex: 232-2024 YHPTOK
A,CH,CM,CS,E,MP,P*

Yokogawa-Hewlett-Packard Ltd.
Daiichi Asano Building
2-8, Odori, 5 Chome
UTSUNOMIYA, Tochigi 320
Tel: (0286) 25-7155
CH,CS,E

Yokogawa-Hewlett-Packard Ltd.
Yasuda Seimei Nishiguchi Bldg.
30-4 Tsuruya-cho, 3 Chome
YOKOHAMA 221
Tel: (045) 312-1252
CH,CM,E

JORDAN
Mouasher Cousins Company
P.O. Box 1387
AMMAN
Tel: 24907, 39907
Telex: 21456 SABCO JO
CH,E,M,P

KENYA
ADCOM Lld., Inc., Kenya
P.O.Box 30070
NAIROBI
Tel: 331955
Telex: 22639
E,M

KOREA
Samsung Electronics HP Division
12 Fl. Kinam Bldg.
San 75-31, Yeoksam-Dong
Kangnam-Ku
Yeongdong P.O. Box 72
SEOUL
Tel: 555-7555, 555-5447
Telex: K27364 SAMSAN
A,CH,CM,CS,E,M,P

KUWAIT
Al-Khaldiya Trading & Conlracling
P.O. Box 830 Safal
KUWAIT
Tel: 42-4910, 41-1726
Telex: 22481 Areeg kl
CH,E,M

Photo & Cine Equipment
P.O. Box 270 Safal
KUWAIT
Tel: 42-2846, 42-3801
Telex: 22247 Malin kt
P

LEBANON
G.M. Dolmadjian
Achrafieh
P.O. Box 165.167
BEIRUT
Tel: 290293
MP**
Computer Information Syslems
P.O. Box 11-6274
BEIRUT
Tel: 89 40 73
Telex: 22259
C

LUXEMBOURG
Hewlett-Packard Belgium S.A./N.V.
Blvd de la Woluwe, 100
Woluwedal
B-1200 **BRUSSELS**
Tel: (02) 762-32-00
Telex: 23-494 paloben bru
A,CH,CM,CS,E,MP,P

MALAYSIA
Hewlett-Packard Sales (Malaysia)
Sdn. Bhd.
1st Floor, Bangunan British
American
Jalan Semantan, Oamansara Heights
KUALA LUMPUR 23-03
Tel: 943022
Telex: MA31011
A,CH,E,M,P*

MAYLAYSIA (Cont'd)
Protel Engineering
P.O.8ox 1917
Lot 6624, Section 64
23/4 Pending Road
Kuching, SARAWAK
Tel: 36299
Telex: MA 70904 PROMAL
Cable: PROTELENG
A,E,M

MALTA
Philip Toledo Ltd.
Notabile Rd.
MRIEHEL
Tel: 447 47, 455 66
Telex: Media MW 649
E,P

MEXICO
Hewlett-Packard Mexicana, S.A.
de C.V.
Av. Periferico Sur No. 6501
Tepepan, Xochimilco
16020 MEXICO D.F.
Tel: 6-76-46-00
Telex: 17-74-507 HEWPACK MEX
A,CH,CS,E,MS,P

Hewlett-Packard Mexicana, S.A.
de C.V.
Ave. Colonia del Valle 409
Col. del Valle
Municipio de Garza Garcia
MONTERREY, Nuevo Leon
Tel: 78 42 41
Telex: 038 410
CH

ECISA
José Vasconcelos No. 218
Col. Condesa Deleg. Cuauhtémoc
MEXICO D.F. 06140
Tel: 553-1206
Telex: 17-72755 ECE ME
M

MOROCCO
Dolbeau
81 rue Karatchi
CASABLANCA
Tel: 3041-82, 3068-38
Telex: 23051, 22822
E

Gerep
2 rue d'Agadir
Boite Postale 156
CASABLANCA
Tel: 272093, 272095
Telex: 23 739
P

NETHERLANDS
Hewlett-Packard Nederland B.V.
Van Heuven Goedhartlaan 121
NL 1181KK AMSTELVEEN
P.O. Box 667
NL1180 AR AMSTELVEEN
Tel: (020) 47-20-21
Telex: 13 216 HEPA NL
A,CH,CM,CS,E,MP,P

Hewlett-Packard Nederland B.V.
Bongerd 2
NL 2906VK CAPELLE A/D IJSSEL
P.O. Box 41
NL 2900AA CAPELLE A/D IJSSEL
Tel: (10) 51-64-44
Telex: 21261 HEPAC NL
A,CH,CS,E

Hewlett-Packard Nederland B.V.
Pastoor Petersstraat 134-136
NL 5612 LV EINDHOVEN
P.O. Box 2342
NL 5600 CH EtNDHOVEN
Tel: (040) 326911
Telex: 51484 hepae nl
A,CH**,E,M

NEW ZEALAND
Hewlett-Packard (N.Z.) Ltd.
5 Owens Road
P.O. Box 26-189
Epsom, AUCKLAND
Tel: 687-159
Cable: HEWPACK Auckland
CH,CM,E,P*

Hewlett-Packard (N.Z.) Ltd.
4-12 Cruickshank Street
Kilbirnie, WELLINGTON 3
P.O. Box 9443
Courtenay Place, WELLINGTON 3
Tel: 877-199
Cable: HEWPACK Wellington
CH,CM,E,P

Northrop Instruments & Systems Ltd.
369 Khyber Pass Road
P.O. Box 8602
AUCKLAND
Tel: 794-091
Telex: 60605
A,M

Northrop Instruments & Systems Ltd.
110 Mandeville St.
P.O. 8ox 8388
CHRISTCHURCH
Tel: 486-928
Telex: 4203
A,M

Northrop Instruments & Systems Ltd.
Sturdee House
85-87 Ghuznee Street
P.O. 8ox 2406
WELLINGTON
Tel: 850-091
Telex: NZ 3380
A,M

NORTHERN IRELAND
See United Kingdom

NORWAY
Hewlett-Packard Norge A/S
Folke Bernadottes vei 50
P.O. Box 3558
N-5033 FYLLINGSDALEN (Bergen)
Tel: 0047/5/16 55 40
Telex: 16621 hpnas n
CH,CS,E,MS

Hewlett-Packard Norge A/S
Österndalen 16-18
P.O. Box 34
N-1345 ÖSTERÅS
Tel: 0047/2/17 11 80
Telex: 16621 hpnas n
A,CH,CM,CS,E,M,P

OMAN
Khimjil Ramdas
P.O. 8ox 19
MUSCAT
Tel: 722225, 745601
Telex: 3289 8ROKER MB MUSCAT
P

Suhail & Saud 8ahwan
P.O.Box 169
MUSCAT
Tel: 734 201-3
Telex: 3274 8AHWAN M8

PAKISTAN
Mushko & Company Ltd.
1-B, Street 43
Sector F-8/1
ISLAMABAD
Tel: 51071
Cable: FEMUS Rawalpindi
A,E,M

Mushko & Company Ltd.
Oosman Chambers
Abdullah Haroon Road
KARACHt 0302
Tel: 524131, 524132
Telex: 2894 MUSKO PK
Cable: COOPERATOR Karachi
*A,E,M,P**

PANAMA
Electrónico Balboa, S.A.
Calle Samuel Lewis, Ed. Alfa
Apartado 4929
PANAMA 5
Tel: 63-6613, 63-6748
Telex: 3483 ELECTRON PG
A,CM,E,M,P

PERU
Cía Electro Médica S.A.
Los Flamencos 145, San Isidro
Casilla 1030
LIMA 1
Tel: 41-4325, 41-3703
Telex: Pub. Booth 25306
CM,E,M,P

PHILIPPINES
The Online Advanced Systems
Corporation
Rico House, Amorsolo Cor. Herrera
Street
Legaspi Village, Makati
P.O. 8ox 1510
Metro MANILA
Tel: 85-35-81, 85-34-91, 85-32-21
Telex: 3274 ONLINE
A,CH,CS,E,M

Electronic Specialists and Proponents
Inc.
690-8 Epifanio de los Santos Avenue
Cubao, QUEZON CITY
P.O. 8ox 2649 Manila
Tel: 98-96-81, 98-96-82, 98-96-83
Telex: 40018, 42000 ITT GLOBE
MACKAY 8OOTH
P

PORTUGAL
Mundinter
Intercambio Mundial de Comércio
S.A.R.L.
P.O. Box 2761
Av. Antonio Augusto de Aguiar 138
P-LISBON
Tel: (19) 53-21-31, 53-21-37
Telex: 16691 munter p
M

Soquimica
Av. da Liberdade, 220-2
1298 LISBOA Codex
Tel: 56 21 81/2/3
Telex: 13316 SA8ASA
P

Telectra-Empresa Técnica de
Equipmentos Eléctricos S.A.R.L.
Rua Rodrigo da Fonseca 103
P.O. Box 2531
P-LISBON 1
Tel: (19) 68-60-72
Telex: 12598
CH,CS,E,P

PUERTO RICO
Hewlett-Packard Puerto Rico
Ave. Muñoz Rivera #101
Esq. Calle Ochoa
HATO REY, Puerto Rico 00918
Tel: (809) 754-7800

Hewlett-Packard Puerto Rico
Calle 272 Edificio 203
Urb. Country Club
RIO PIEDRAS, Puerto Rico
P.O. Box 4407
CAROLINA, Puerto Rico 00628
Tel: (809) 762-7255
A,CH,CS

QATAR
Computearbia
P.O. 8ox 2750
DOHA
Tel: 883555
Telex: 4806 CHPARB
P

Eastern Technical Services
P.O.8ox 4747
DOHA
Tel: 329 993
Telex: 4156 EASTEC DH

Nasser Trading & Contracting
P.O.Box 1563
DOHA
Tel: 22170, 23539
Telex: 4439 NASSER DH
M

SAUDI ARABIA
Modern Electronic Establishment
Hewlett-Packard Division
P.O. Box 22015
Thuobah
AL-KHOBAR
Tel: 895-1760, 895-1764
Telex: 671 106 HPMEEK SJ
Cable: ELECTA AL-KHO8AR
CH,CS,E,M

Modern Electronic Establishment
Hewlett-Packard Division
P.O. 8ox 1228
Redec Plaza, 6th Floor
JEDDAH
Tel: 644 38 48
Telex: 4027 12 FARNAS SJ
Cable: ELECTA JEDDAH
CH,CS,E,M

Modern Electronic Establishment
Hewlett-Packard Division
P.O.8ox 22015
RtYADH
Tel: 491-97 15, 491-63 87
Telex: 202049 MEERYD SJ
CH,CS,E,M

Abdul Ghani El Ajou
P.O. Box 78
RtYADH
Tel: 40 41 717
Telex: 200 932 EL AJOU
P

SCOTLAND
See United Kingdom

SINGAPORE
Hewlett-Packard Singapore (Sales)
Pte. Ltd.
#08-00 Inchcape House
450-2 Alexandra Road
P.O. Box 58 Alexandra Rd. Post Office
SINGAPORE, 9115
Tel: 631788
Telex: HPSGSO RS 34209
Cable: HEWPACK, Singapore
A,CH,CS,E,MS,P

SINGAPORE (Cont'd)
Dynamar International Ltd.
Unit 05-11 Block 6
Kolam Ayer Industrial Estate
SINGAPORE 1334
Tel: 747-6188
Telex: RS 26283
CM

SOUTH AFRICA
Hewlett-Packard So Africa (Pty.) Ltd.
P.O. Box 120
Howard Place **CAPE PROVINCE 7450**
Pine Park Center, Forest Drive,
Pinelands
CAPE PROVINCE 7405
Tel: 53-7954
Telex: 57-20006
A,CH,CM,E,MS,P

Hewlett-Packard So Africa (Pty.) Ltd.
P.O. Box 37099
92 Overport Drive
DURBAN 4067
Tel: 28-4178, 28-4179, 28-4110
Telex: 6-22954
CH,CM

Hewlett-Packard So Africa (Pty.) Ltd.
6 Linton Arcade
511 Cape Road
Linton Grange
PORT ELIZABETH 6000
Tel: 041-302148
CH

Hewlett-Packard So Africa (Pty.) Ltd.
P.O.Box 33345
Glenstantia 0010 **TRANSVAAL**
1st Floor East
Constantia Park Ridge Shopping
Centre
Constantia Park
PRETORIA
Tel: 982043
Telex: 32163
CH,E

Hewlett-Packard So Africa (Pty.) Ltd.
Private Bag Wendywood
SANDTON 2144
Tel: 802-5111, 802-5125
Telex: 4-20877
Cable: HEWPACK Johannesburg
A,CH,CM,CS,E,MS,P

SPAIN
Hewlett-Packard Española S.A.
Calle Entenza, 321
E-BARCELONA 29
Tel: 322.24.51, 321.73.54
Telex: 52603 hpbee
A,CH,CS,E,MS,P

Hewlett-Packard Española S.A.
Calle San Vicente S/No
Edificio Albia II
E-BILBAO 1
Tel: 423.83.06
A,CH,E,MS

Hewlett-Packard Española S.A.
Crta. de la Coruña, Km. 16, 400
Las Rozas
E-MADRID
Tel: (1) 637.00.11
CH,CS,M

Hewlett-Packard Española S.A.
Avda. S. Francisco Javier, S/no
Planta 10. Edificio Sevilla 2,
E-SEVILLA 5
Tel: 64.44.54
Telex: 72933
A,CS,MS,P

Hewlett-Packard Española S.A.
Calle Ramon Gordillo, 1 (Entlo.3)
E-VALENCIA 10
Tel: 361-1354
CH,P

SWEDEN
Hewlett-Packard Sverige AB
Sunnanvagen 14K
S-22226 LUND
Tel: (046) 13-69-79
Telex: (854) 17886 (via Spånga
office)
CH

Hewlett-Packard Sverige AB
Östra Tullgatan 3
S-21128 MALMÖ
Tel: (040) 70270
Telex: (854) 17886 (via Spånga
office)

Hewlett-Packard Sverige AB
Våstra Vintergatan 9
S-70344 ÖREBRO
Tel: (19) 10-48-80
Telex: (854) 17886 (via Spånga
office)
CH

Hewlett-Packard Sverige AB
Skalholtsgatan 9, Kista
Box 19
S-16393 SPÅNGA
Tel: (08) 750-2000
Telex: (854) 17886
Telefax: (08) 7527781
A,CH,CM,CS,E,MS,P

Hewlett-Packard Sverige AB
Frötallisgatan 30
S-42132 VÄSTRA-FRÖLUNDA
Tel: (031) 49-09-50
Telex: (854) 17886 (via Spånga
office)
CH,E,P

SWITZERLAND
Hewlett-Packard (Schweiz) AG
Clarastrasse 12
CH-4058 BASEL
Tel: (61) 33-59-20
A

Hewlett-Packard (Schweiz) AG
7, rue du Bois-du-Lan
Case Postale 365
CH-1217 MEYRIN 2
Tel: (0041) 22-83-11-11
Telex:27333 HPAG CH
CH,CM,CS

Hewlett-Packard (Schweiz) AG
Allmend 2
CH-8967 WIDEN
Tel: (0041) 57 31 21 11
Telex: 53933 hpag ch
Cable: HPAG CH
A,CH,CM,CS,E,MS,P

SYRIA
General Electronic Inc.
Nuri Basha Ahnaf Ebn Kays Street
P.O. Box 5781
DAMASCUS
Tel: 33-24-87
Telex: 411 215
Cable: ELECTROBOR DAMASCUS
E

Middle East Electronics
P.O.Box 2308
Abu Rumnaneh
DAMASCUS
Tel: 33 4 5 92
Telex: 411 304
M

TAIWAN
Hewlett-Packard Far East Ltd.
Kaohsiung Office
2/F 68-2, Chung Cheng 3rd Road
KAOHSIUNG
Tel: (07) 241-2318
CH,CS,E

Hewlett-Packard Far East Ltd.
Taiwan Branch
8th Floor
337 Fu Hsing North Road
TAIPEI
Tel: (02) 712-0404
Telex: 24439 HEWPACK
Cable:HEWPACK Taipei
A,CH,CM,CS,E,M,P

Ing Lih Trading Co.
3rd Floor, 7 Jen-Ai Road, Sec. 2
TAIPEI 100
Tel: (02) 3948191
Cable: INGLIH TAIPEI
A

THAILAND
Unimesa
30 Patpong Ave., Suriwong
BANGKOK 5
Tel: 235-5727
Telex: 84439 Simonco TH
Cable: UNIMESA Bangkok
A,CH,CS,E,M

Bangkok Business Equipment Ltd.
5/5-6 Dejo Road
BANGKOK
Tel: 234-8670, 234-8671
Telex: 87669-BEQUIPT TH
Cable: BUSIQUIPT Bangkok
P

TRINIDAD & TOBAGO
Caribbean Telecoms Ltd.
50/A Jerningham Avenue
P.O. Box 732
PORT-OF-SPAIN
Tel: 62-44213, 62-44214
Telex: 235,272 HUGCO WG
CM,E,M,P

TUNISIA
Tunisie Electronique
31 Avenue de la Liberte
TUNIS
Tel: 280-144
E,P

Corema
1 ler. Av. de Carthage
TUNIS
Tel: 253-821
Telex: 12319 CABAM TN
M

TURKEY
Teknim Company Ltd.
Iran Caddesi No. 7
Kavaklidere, ANKARA
Tel: 275800
Telex: 42155 TKNM TR
E

E.M.A.
Medina Eldem Sokak No.41/6
Yuksel Caddesi
ANKARA
Tel: 175 622
Telex: 42 591
M

UNITED ARAB EMIRATES
Emitac Ltd.
P.O. Box 2711
ABU DHABI
Tel: 82 04 19-20
Cable: EMITAC ABUDHABI

Emitac Ltd.
P.O. Box 1641
SHARJAH
Tel: 591 181
Telex: 68136 Emitac Sh
CH,CS,E,M,P

UNITED KINGDOM
GREAT BRITAIN
Hewlett-Packard Ltd.
Trafalgar House
Navigation Road
ALTRINCHAM
Cheshire WA14 1NU
Tel: 061 928 6422
Telex: 668068
A,CH,CS,E,M,MS,P

Hewlett-Packard Ltd.
Elstree House, Elstree Way
BOREHAMWOOD, Herts WD6 1SG
Tel: 01 207 5000
Telex: 8952716
E,CH,CS,P

Hewlett-Packard Ltd.
Oakfield House, Oakfield Grove
Clifton **BRISTOL**, Avon BS8 2BN
Tel: 0272 736806
Telex: 444302
CH,CS,E,P

Hewlett-Packard Ltd.
Bridewell House
Bridewell Place
LONDON EC4V 6BS
Tel: 01 583 6565
Telex: 298163
CH,CS,P

Hewlett-Packard Ltd.
Fourier House
257-263 High Street
LONDON COLNEY
Herts. AL2 1HA, St. Albans
Tel: 0727 24400
Telex: 1-8952716
CH,CS

Hewlett-Packard Ltd.
Pontefract Road
NORMANTON, West Yorkshire WF6 1RN
Tel: 0924 895566
Telex: 557355
CH,CS,P

Hewlett-Packard Ltd.
The Quadrangle
106-118 Station Road
REDHILL, Surrey RH1 1PS
Tel: 0737 68655
Telex: 947234
CH,CS,E,P

GREAT BRITAIN (Cont'd)
Hewlett-Packard Ltd.
Avon House
435 Stratford Road
Shirley, **SOLIHULL**, West Midlands
B90 4BL
Tel: 021 745 8800
Telex: 339105
CH,CS,E,P

Hewlett-Packard Ltd.
West End House
41 High Street, West End
SOUTHAMPTON
Hampshire S03 3DQ
Tel: 04218 6767
Telex: 477138
CH,CS,P

Hewlett-Packard Ltd.
Eskdale Rd.
Winnersh, **WOKINGHAM**
Berkshire RG11 50Z
Tel: 0734 696622
Telex: 848884
E

Hewlett-Packard Ltd.
King Street Lane
Winnersh, **WOKINGHAM**
Berkshire RG11 5AR
Tel: 0734 784774
Telex: 847178
A,CH,CS,E,M,MP,P

Hewlett-Packard Ltd.
Nine Mile Ride
Easthampstead, **WOKINGHAM**
Berkshire, 3RG11 3LL
Tel: 0344 773100
Telex: 848805
CH,CS,E,P

IRELAND

NORTHERN IRELAND
Hewlett-Packard Ltd.
Cardiac Services Building
95A Finaghy Road South
BELFAST BT10 0BY
Tel: 0232 625-566
Telex: 747626
CH,CS

SCOTLAND
Hewlett-Packard Ltd.
SOUTH QUEENSFERRY
West Lothian, EH30 9TG
Tel: 031 331 1188
Telex: 72682
CH,CM,CS,E,M,P

UNITED STATES

Alabama
Hewlett-Packard Co.
700 Century Park South, Suite 128
BIRMINGHAM, AL 35226
Tel: (205) 822-6802
A,CH,M

Hewlett-Packard Co.
420 Wynn Drive
HUNTSVILLE, AL 35805
P.O. Box 7700
HUNTSVILLE, AL 35807
Tel: (205) 830-2000
CH,CM,CS,E,M*

Arizona
Hewlett-Packard Co.
8080 Pointe Parkway West
PHOENIX, AZ 85044
Tel: (602) 273-8000
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
2424 East Aragon Road
TUCSON, AZ 85706
Tel: (602) 889-4631
CH,E,MS**

California
Hewlett-Packard Co.
99 South Hill Or.
BRISBANE, CA 94005
Tel: (415) 330-2500
CH,CS

Hewlett-Packard Co.
P.O. Box 7830 (93747)
5060 E. Clinton Avenue, Suite 102
FRESNO, CA 93727
Tel: (209) 252-9652
CH,CS,MS

Hewlett-Packard Co.
P.O. Box 4230
1430 East Orangethorpe
FULLERTON, CA 92631
Tel: (714) 870-1000
CH,CM,CS,E,MP

Hewlett-Packard Co.
320 S. Kellogg, Suite B
GOLETA, CA 93117
Tel: (805) 967-3405
CH

Hewlett-Packard Co.
5400 W. Rosecrans Boulevard
LAWNDALE, CA 90260
P.O. Box 92105
LOS ANGELES, CA 90009
Tel: (213) 970-7500
Telex: 910-325-6608
CH,CM,CS,MP

Hewlett-Packard Co.
3155 Porter Oaks Drive
PALO ALTO, CA 94304
Tel: (415) 857-8000
CH,CS,E

Hewlett-Packard Co.
4244 So. Market Court, Suite A
P.O. Box 15976
SACRAMENTO, CA 95852
Tel: (916) 929-7222
A*,CH,CS,E,MS

Hewlett-Packard Co.
9606 Aero Orive
P.O. Box 23333
SAN DIEGO, CA 92139
Tel: (619) 279-3200
CH,CM,CS,E,MP

Hewlett-Packard Co.
2305 Camino Ramon "C"
SAN RAMON, CA 94583
Tel: (415) 838-5900
CH,CS

Hewlett-Packard Co.
3005 Scott Boulevard
SANTA CLARA, CA 95050
Tel: (408) 988-7000
Telex: 910-338-0586
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
5703 Corsa Avenue
WESTLAKE VILLAGE, CA 91362
Tel: (213) 706-6800
E*,CH*,CS*

Colorado
Hewlett-Packard Co.
24 Inverness Place, East
ENGLEWOOD, CO 80112
Tel: (303) 649-5000
A,CH,CM,CS,E,MS

Connecticut
Hewlett-Packard Co.
47 Barnes Industrial Road South
P.O. Box 5007
WALLINGFORD, CT 06492
Tel: (203) 265-7801
A,CH,CM,CS,E,MS

Florida
Hewlett-Packard Co.
2901 N.W. 62nd Street
P.O. Box 24210
FORT LAUDERDALE, FL 33307
Tel: (305) 973-2600
CH,CS,E,MP

Hewlett-Packard Co.
6177 Lake Ellenor Orive
P.O. Box 13910
ORLANDO, FL 32859
Tel: (305) 859-2900
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
5750B N. Hoover Blvd., Suite 123
P.O. Box 15200
TAMPA, FL 33614
Tel: (813) 884-3282
A*,CH,CM,CS,E*,M*

Georgia
Hewlett-Packard Co.
2000 South Park Place
P.O. Box 105005
ATLANTA, GA 30348
Tel: (404) 955-1500
Telex: 810-766-4890
A,CH,CM,CS,E,MP

Hawaii
Hewlett-Packard Co.
Kawaiahao Plaza, Suite 190
567 South King Street
HONOLULU, HI 96813
Tel: (808) 526-1555
A,CH,E,MS

Illinois
Hewlett-Packard Co.
304 Eldorado Road
P.O. Box 1607
BLOOMINGTON, IL 61701
Tel: (309) 662-9411
CH,MS**

Hewlett-Packard Co.
1100 31st Street, Suite 100
DOWNERS GROVE, IL 60515
Tel: (312) 960-5760
CH,CS

Hewlett-Packard Co.
5201 Tollview Orive
ROLLING MEADOWS, IL 60008
Tel: (312) 255-9800
Telex: 910-687-1066
A,CH,CM,CS,E,MP

Indiana
Hewlett-Packard Co.
7301 No. Shadeland Avenue
P.O. Box 50807
INDIANAPOLIS, IN 46250
Tel: (317) 842-1000
A,CH,CM,CS,E,MS

Iowa
Hewlett-Packard Co.
1776 22nd Street, Suite 1
WEST DES MOINES, IA 50265
Tel: (515) 224-1435
CH,MS**

Kansas
Hewlett-Packard Co.
7804 East Funston Road, #203
WICHITA, KS 67207
Tel: (316) 684-8491
CH

Kentucky
Hewlett-Packard Co.
10300 Linn Station Road, #100
LOUISVILLE, KY 40223
Tel: (502) 426-0100
A,CH,CS,MS

Louisiana
Hewlett-Packard Co.
160 James Orive East
ST. ROSE, LA 70087
P.O. Box 1449
KENNER, LA 70063
Tel: (504) 467-4100
A,CH,CS,E,MS

Maryland
Hewlett-Packard Co.
3701 Koppers Street
BALTIMORE, MO 21227
Tel: (301) 644-5800
Telex: 710-862-1943
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
2 Choke Cherry Road
ROCKVILLE, MO 20850
Tel: (301) 948-6370
A,CH,CM,CS,E,MP

Massachusetts
Hewlett-Packard Co.
1775 Minuteman Road
ANDOVER, MA 01810
Tel: (617) 682-1500
A,C,CH,CS,CM,E,MP,P*

Hewlett-Packard Co.
32 Hartwell Avenue
LEXINGTON, MA 02173
Tel: (617) 861-8960
CH,CS,E

Michigan
Hewlett-Packard Co.
4326 Cascade Road S.E.
GRAND RAPIDS, MI 49506
Tel: (616) 957-1970
CH,CS,MS

Hewlett-Packard Co.
1771 W. Big Beaver Road
TROY, MI 48084
Tel: (313) 643-6474
CH,CS

Minnesota
Hewlett-Packard Co.
2025 W. Larpenteur Ave.
ST. PAUL, MN 55113
Tel: (612) 644-1100
A,CH,CM,CS,E,MP

Missouri
Hewlett-Packard Co.
11131 Colorado Avenue
KANSAS CITY, MO 64137
Tel: (816) 763-8000
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
13001 Hollenberg Orive
BRIDGETON, MO 63044
Tel: (314) 344-5100
A,CH,CS,E,MP

UNITED STATES (Cont'd)

Nebraska
Hewlett-Packard
10824 Old Mill Rd., Suite 3
OMAHA, NE 68154
Tel: (402) 334-1813
CM,MS

New Jersey
Hewlett-Packard Co.
120 W. Century Road
PARAMUS, NJ 07652
Tel: (201) 265-5000
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
60 New England Av. West
PISCATAWAY, NJ 08854
Tel: (201) 981-1199
A,CH,CM,CS,E

New Mexico
Hewlett-Packard Co.
11300 Lomas Blvd.,N.E.
P.O. Box 11634
ALBUQUERQUE, NM 87112
Tel: (505) 292-1330
CH,CS,E,MS

New York
Hewlett-Packard Co.
5 Computer Drive South
ALBANY, NY 12205
Tel: (518) 458-1550
A,CH,E,MS

Hewlett-Packard Co.
9600 Main Street
P.O. Box AC
CLARENCE, NY 14031
Tel: (716) 759-8621
CH

Hewlett-Packard Co.
200 Cross Keys Office Park
FAIRPORT, NY 14450
Tel: (716) 223-9950
CH,CM,CS,E,MS

Hewlett-Packard Co.
7641 Henry Clay Blvd.
LIVERPOOL, NY 13088
Tel: (315) 451-1820
A,CH,CM,E,MS

Hewlett-Packard Co.
No. 1 Pennsylvania Plaza
55th Floor
34th Street & 8th Avenue
MANHATTAN NY 10119
Tel: (212) 971-0800
CH,CS,E*,M*

Hewlett-Packard Co.
250 Westchester Avenue
WHITE PLAINS, NY 10604
Tel: (914) 684-6100
CM,CH,CS,E

Hewlett-Packard Co.
3 Crossways Park West
WOODBURY, NY 11797
Tel: (516) 921-0300
A,CH,CM,CS,E,MS

North Carolina
Hewlett-Packard Co.
5605 Roanne Way
P.O. Box 26500
GREENSBORO, NC 27420
Tel: (919) 852-1800
A,CH,CM,CS,E,MS

Ohio
Hewlett-Packard Co.
9920 Carver Road
CINCINNATI, OH 45242
Tel: (513) 891-9870
CH,CS,MS

Hewlett-Packard Co.
16500 Sprague Road
CLEVELAND, OH 44130
Tel: (216) 243-7300
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
962 Crupper Ave.
COLUMBUS, OH 43229
Tel: (614) 436-1041
Eff: Nov. 25, 1983
675 Brooksedge Blvd.
WESTERVILLE, OH 43081
CH,CM,CS,E*

Hewlett-Packard Co.
330 Progress Rd.
DAYTON, OH 45449
Tel: (513) 859-8202
A,CH,CM,E*,MS

Oklahoma
Hewlett-Packard Co.
304 N. Meridian, Suite A
P.O. Box 75609
OKLAHOMA CITY, OK 73147
Tel: (405) 946-9499
A*,CH,E*,MS

Hewlett-Packard Co.
3840 S. 103rd E. Avenue, #100
P.O. Box 35747
TULSA, OK 74153
Tel: (918) 665-3300
A**,CH,CS,M*

Oregon
Hewlett-Packard Co.
9255 S. W. Pioneer Court
P.O. Box 328
WILSONVILLE, OR 97070
Tel: (503) 682-8000
A,CH,CS,E*,MS

Pennsylvania
Hewlett-Packard Co.
111 Zeta Drive
PITTSBURGH, PA 15238
Tel: (412) 782-0400
A,CH,CS,E,MP

Hewlett-Packard Co.
2750 Monroe Boulevard
P.O. Box 713
VALLEY FORGE, PA 19482
Tel: (215) 666-9000
A,CH,CM,E,M

South Carolina
Hewlett-Packard Co.
Brookside Park, Suite 122
1 Harbison Way
P.O. Box 21708
COLUMBIA, SC 29221
Tel: (803) 732-0400
CH,E,MS

Hewlett-Packard Co.
Koger Executive Center
Chesterfield Bldg., Suite 124
GREENVILLE, SC 29615
Tel: (803) 297-4120

Tennessee
Hewlett-Packard Co.
224 Peters Road, Suite 102
P.O. Box 22490
KNOXVILLE, TN 37922
Tel: (615) 691-2371
A*,CH,MS

Hewlett-Packard Co.
3070 Directors Row
MEMPHIS, TN 38131
Tel: (901) 346-8370
A,CH,MS

Texas
Hewlett-Packard Co.
4171 North Mesa
Suite C-110
EL PASO, TX 79902
Tel: (915) 533-3555
CH,E*,MS**

Hewlett-Packard Co.
10535 Harwin Drive
P.O. Box 42816
HOUSTON, TX 77042
Tel: (713) 776-6400
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
930 E. Campbell Rd.
P.O. Box 1270
RICHARDSON, TX 75080
Tel: (214) 231-6101
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
1020 Central Parkway South
P.O. Box 32993
SAN ANTONIO, TX 78216
Tel: (512) 494-9336
CH,CS,E,MS

Utah
Hewlett-Packard Co.
3530 W. 2100 South
SALT LAKE CITY, UT 84119
Tel: (801) 974-1700
A,CH,CS,E,MS

Virginia
Hewlett-Packard Co.
4305 Cox Road
GLEN ALLEN, VA 23060
P.O. Box 9669
RICHMOND, VA 23228
Tel: (804) 747-7750
A,CH,CS,E,MS

Washington
Hewlett-Packard Co.
15815 S.E. 37th Street
BELLEVUE, WA 98006
Tel: (206) 643-4000
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
Suite A
708 North Argonne Road
SPOKANE, WA 99212
Tel: (509) 922-7000
CH,CS

West Virginia
Hewlett-Packard Co.
4604 MacCorkle Ave.
P.O. Box 4297
CHARLESTON, WV 25304
Tel: (304) 925-0492
A,MS

Wisconsin
Hewlett-Packard Co.
150 S. Sunny Slope Road
BROOKFIELD, WI 53005
Tel: (414) 784-8800
A,CH,CS,E*,MP

URUGUAY
Pablo Ferrando S.A.C. e I.
Avenida Italia 2877
Casilla de Correo 370
MONTEVIDEO
Tel: 80-2586
Telex: Public Booth 901
A,CM,E,M

VENEZUELA
Hewlett-Packard de Venezuela C.A.
3RA Transversal Los Ruices Norte
Edificio Segre 1, 2 & 3
Apartado 50933
CARACAS 1071
Tel: 239-4133
Telex: 251046 HEWPACK
A,CH,CS,E,MS,P

Hewlett-Packard de Venezuela C.A.
Calle-72-Entre 3H y 3Y, No. 3H-40
Edificio Ada-Evelyn, Local 8
Apartado 2646
4001, MARACAIBO, Estado Zulia
Tel: (061) 80.304
C,E*

Hewlett-Packard de Venezuela C.A.
Calle Vargas Rondon
Edificio Seguros Carabobo, Piso 10
VALENCIA
Tel: (041) 51 385
CH,CS,P

Bioelectronica Medica C.A.
Calle Buen Pastor
Edif. Cota Mil-Piso 2 y Semi Sotano 1
Boleita Norte
Apartado 50710 CARACAS 1050A
Tel: 239 84 41
Telex: 26518

ZIMBABWE
Field Technical Sales
45 Kelvin Road, North
P.B. 3458
SALISBURY
Tel: 705 231
Telex: 4-122 RH
C,E,M,P

July 1983   5952-6900

Indicates main office

HP distributors are printed in italics.

HEWLETT
PACKARD